

MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

MINIX 3 FREE - OPEN-SOURCE

OPERATING SYSTEM, HIGHLY RELIABLE, FLEXIBLE, AND SECURE

MINIX

A CLASS-BASED OPERATING SYSTEM

DEPLOY DOCKER SWARM CLUSTER ON ONE HOST

FREE NAS GETTING STARTED GUIDE:

PART 3, MANUAL CONFIGURATION

OPTIMIZING IN-MEMORY

CACHE OF THE BEAST ARCHITECTURE

USER STORY

FROM THE OO ARCHITECTURE POINT OF VIEW

FIXING FAILING PORTS

FOR HARDENED/LIBREBSD

VOL 10 NO 08

ISSUE 07/2016 (84)

1898-9144

THE FREENAS MINI XL HAS ARRIVED



ENTERPRISE-CLASS HARDWARE, RUNNING
THE WORLD'S MOST POPULAR OPEN SOURCE
STORAGE OPERATING SYSTEM.

For more information on the FreeNAS Mini,
visit ixsystems.com/mini today.

Dear Readers,

The summer holiday season is almost behind us. We hope you have been having a great time. At BSD Mag, we have been working hard to serve you the content that will interest and benefit you the most. We have sent a couple of requests for you to let us know what topics have been the most interesting for you. We will try to find articles on those topics to help you find the answers you are looking for. So, if you have something in mind, don't hesitate to contact us.

We hope that you haven't missed out on the "BSD Special - Best of David Carlier" Issue. If so, go to our web page and download your free copy right away.

Now, let's dive into this issue. Everything about MINIX was one of the topic requests we received from you. That's why, after the News section, you will find two articles about MINIX, which will introduce to you this project: "MINIX 3 - Free, Open Source, Operating System, Highly Reliable, Flexible, and Secure" by Mauro Risonho de Paula Assumpção and "MINIX - A Class-Based Operating System" by Rafael Santiago de Souza Netto.

Next, you will find another great article by Mikhail E. Zakharov, "Optimizing In-Memory Cache of the BeaST Architecture" about The BeaST - the new FreeBSD based, dual-headed, reliable, storage system.

Regarding storage, "Deploy Docker Swarm Cluster on One Host" by Nan Xiao will help you with building a Docker Swarm cluster on one host. This tutorial will provide a detailed guide of the process. We have been very interested in HardenedBSD recently, and you can expect more articles about the project in the near future. In this issue, you will find "Fixing Failing Ports for Hardened/LibreBSD" by Bernard Spil. Let us know if you are interested in more HardenedBSD and LibreBSD articles in upcoming issues.

Great as always, is Mark VonFange and his 3rd part of "FreeNAS Getting Started Guide: Part 3, Manual Configuration". Grab a coffee and get ready for a lot of great content.

From Damian Czernous we have received an introduction to the new series "User Story from the OO Architecture Point of View". Obviously, it's not very BSD-related, so let us know what you think about it and if you would like to see the other parts in upcoming issues.

And in the end, Rob Somerville, as always. Read this month's column about technology that (maybe) went too far.

Marta & BSD Team



Editor in Chief:

Marta Ziemianowicz

marta.ziemianowicz@software.com.pl

Contributing:

Mauro Risonho de Paula Assumpção , Rafael Santiago de Souza Netto, Mikhail E. Zakharov, Bernard Spil, Mark VonFange, Nan Xiao , Kevin McAleer, Damian Czernous and Rob Somerville.

Top Betatesters & Proofreaders:

Annie Zhang, Denise Ebery, Eric Geissinger, Luca Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis Mahangoe, Mani Kanth and Mark VonFange.

Special Thanks:

Annie Zhang

Denise Ebery

DTP:

Marta Ziemianowicz

Senior Consultant/Publisher:

Paweł Marciniak

pawel@software.com.pl

CEO:

Joanna Kretowicz

joanna.kretowicz@software.com.pl

Publisher:

Hakin9 Media SK 02-676 Warsaw, Poland Postepu 17D Poland worldwide publishing editors@bsdmag.org www.bsdmag.org

Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

All trademarks presented in the magazine were used only for informative purposes. All rights to trademarks presented in the magazine are reserved by the companies which own them.

News

BSD World Monthly News **5**

by Marta Ziemianowicz

This column presents the latest news coverage of events, product releases and trending topics.

MINIX

MINIX 3 Free, Open-Source, Operating System, Highly Reliable, Flexible, and Secure. **16**

by Mauro Risonho de Paula Assumpção

MINIX 3 was publicly announced on 24 October 2005 by Andrew Tanenbaum during his keynote speech at the ACM Symposium Operating Systems Principles conference. Although it still serves as an example for the new edition of Tanenbaum and Woodhull's textbook, it is comprehensively redesigned to be "usable as a serious system on resource-limited and embedded computers and for applications requiring high reliability."

MINIX - A Class-Based Operating System **47**

by Rafael Santiago de Souza Netto

This first article intends to introduce the MINIX Operating System, as well as talk about some basic technical and historical aspects involved with it. Also, it will include some general details about MINIX. In addition, you will learn more about Operating Systems in general.

FreeBSD

Optimizing In-Memory Cache of the BeaST Architecture **54**

by Mikhail E. Zakharov

The BeaST is the new FreeBSD based dual-headed reliable storage system concept. Recently, we implemented both ZFS and in-memory cache in our architecture. After this last improvement, the BeaST system has become quite complex compared to its predecessors.

HardenedBSD

Fixing Failing Ports for Hardened/LibreBSD **68**

by Bernard Spil

HardenedBSD ran an exp-run with LibreSSL in base. This was expected to uncover a lot of issues where ports check the OPENSSL_VERSION_NUMBER to determine if a feature is available. To my surprise, it only uncovered 12 ports that failed due to these version checks.

Docker

Deploy Docker Swarm Cluster on One Host **74**

by Nan Xiao

Sometimes, you just want to learn the internal mechanics of Docker Swarm, but, unfortunately, there is only one Linux box at hand, and you don't want to bother to install Virtual Machines on it. In this scenario, you certainly can build a Docker Swarm cluster on one host, and this tutorial will provide a detailed guide.

ZFS

Using ZFS to Fight Data Rotthe **80**

by Kevin McAleer

Previously, I wrote an article for BigAdmin about why I chose the ZFS file system to ensure my data was safe: "How I Used Solaris OS and ZFS to Solve My Mac OS X Storage Problem."

FreeNAS

FreeNAS Getting Started Guide: Part 3, Manual Configuration **85**

by Mark VonFange

This article series is intended to serve as an introductory guide to assist FreeNAS users in planning, installation, configuration and administration for their FreeNAS storage systems. This month's article will cover basic configuration and administration tasks within the FreeNAS User Interface.

CONTENTS

Architecture

User Story From the OO architecture Point of View **105**

by Damian Czernous

A good user story lays a great foundation for future work and shows engineering awareness of the team. For example, short sentences that follow deductive reasoning (top-down strategy) better corresponds with the way of ensuring object oriented architecture. How? In OO (Object Oriented) architecture, every method works in the context of its class. Every class works in the context of its package, and so on... The good OO architecture forms sentences starting from the top package to the bottom method..

Rob's Column **110**

by Rob Somerville

With current advances in technology and systems, has the sector reached the point of consuming itself?

BSD Certification

The BSD Certification Group Inc. (BSDCG) is a non-profit organization committed to creating and maintaining a global certification standard for system administration on BSD based operating systems.

? WHAT CERTIFICATIONS ARE AVAILABLE?

BSDA: Entry-level certification suited for candidates with a general Unix background and at least six months of experience with BSD systems.

BDSP: Advanced certification for senior system administrators with at least three years of experience on BSD systems. Successful BDSP candidates are able to demonstrate strong to expert skills in BSD Unix system administration.

✓ WHERE CAN I GET CERTIFIED?

We're pleased to announce that after 7 months of negotiations and the work required to make the exam available in a computer based format, that the BSDA exam is now available at several hundred testing centers around the world. Paper based BSDA exams cost \$75 USD. Computer based BSDA exams cost \$150 USD. The price of the BDSP exams are yet to be determined.

Payments are made through our registration website:
<https://register.bsdcertification.org/register/payment>

i WHERE CAN I GET MORE INFORMATION?

More information and links to our mailing lists, LinkedIn groups, and Facebook group are available at our website:
<http://www.bsdcertification.org>

Registration for upcoming exam events is available at our registration website:
<https://register.bsdcertification.org/register/get-a-bsdcg-id>

Raspberry Pi 3's BCM2837 SoC Now Supported By Mainline Linux 4.8



ARM platform enablement continues in Linux 4.8 with several new targets being supported by the mainline Linux kernel. The most notable ARM Linux 4.8 addition is support for the Broadcom SoC used by the Raspberry Pi 3.

The ARM platform code is the latest to add to the new features so far of the Linux 4.8 kernel.

New ARM 32-bit platforms to be supported by Linux 4.8 are Broadcom BCM23550, Freescale i.MX7Solo, Qualcomm MDM9615, and the Renesas r8a7792.

There are fewer 64-bit ARM platforms added this cycle, just: Broadcom BCM2837 and Renesas r8a7796. The BCM2837 is arguably, though, the most interesting addition with that being the SoC used by the Raspberry Pi 3. With this support there, it's looking like the Raspberry Pi 3 is getting closer to be fully supported by an upstream Linux kernel.

In addition to the new platforms, updated platforms worth mentioning include the NVIDIA Tegra X1, Mediatek MT8173, Rockchip RK3399, and ARM Juno. The Tegra X1 work includes USB 3.0, regulars, and display subsystem updates.

http://www.phoronix.com/scan.php?page=news_item&px=ARM-Platforms-Linux-4.8

FreeBSD 11.0-RC1 now available

The fourth BETA build of the 11.0-RELEASE release cycle is now available.

Installer images and memory stick images are available here:

<ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/11.0/>

The image checksums follow at the end of this e-mail.

If you notice problems you can report them through the Bugzilla PR system or on the -stable mailing list.

If you would like to use SVN to do a source based update of an existing system, use the "stable/11" branch.

A summary of changes since 11.0-BETA3 includes:

- The `mtx_trylock_spin(9)` kernel synchronization primitive was added.
- The `machdep.disable_msix_migration` loader tunable has been re-enable for EC2 AMIs.
- The `iwm(4)` and `iwmfw(4)` drivers have been updated.
- The new system hardening options have been fixed to avoid overwriting other options selected during install time.
- Several build-related fixes.
- Several miscellaneous bug fixes.

A list of changes since 10.0-RELEASE are available on the stable/11

Release notes:

<https://www.freebsd.org/relnotes/11-STABLE/relnotes/article.html>

Official announcement:

<https://lists.freebsd.org/pipermail/freebsd-stable/2016-August/085186.html>

Lumina Desktop 1.0 Released

The PC-BSD/TrueOS developers have announced the release of the Lumina Desktop Environment 1.0.

Lumina has been in development for the past four years and this is now the project's first official release. This Qt-based desktop environment is designed to be very customizable, extremely lightweight, and support all modern functionality. Lumina works on not only BSD systems but Linux, too.

http://www.phoronix.com/scan.php?page=news_item&px=Lumina-Desktop-1.0-Released

Features

For Users

- Completely customizable interface! Rather than having to learn how to use a new layout, change the desktop to suit you instead!
- Simple shortcuts for any application! The “favorites” system makes it easy to find and launch applications at any time.
- Extremely lightweight! Allows applications to utilize more of your system hardware and revitalizes older systems!
- Multiple-monitor support! Each monitor is treated as an independent entity – making it great for presentation systems which use a temporary monitor or for workstations which utilize an array of monitors for various tasks.

For System Administrators

- Personalize the initial settings for users with a single configuration file!
- Default applications
- Appearance settings (Theme, Colors, Wallpaper(s), Icons, Fonts, etc)
- Interface layout (desktop icons/plugins, panels, etc)
- Favorite apps/files
- Provides a stable and consistent experience across updates. Cut down on your support time for end-user systems!

For System Builders

- Easily ported to various operating systems (OS), with various optional features setup within a single source file for each OS. Already ported to:
- BSD OS's: TrueOS, FreeBSD, OpenBSD, DragonflyBSD, NetBSD
- Linux OS's: Debian, Gentoo, kFreeBSD, generic "Linux"
- Easily add customized config files for your OS (wallpaper, interface settings, etc).
- Minimal dependencies
- Qt5.2+, Fluxbox, xscreensaver, XCB libraries, other small OS utilities as needed.

<https://lumina-desktop.org/version-1-0-0-released/>

2016 BSDCan Trip Reports

BSDCan 2016 was held at the University of Ottawa in Ontario on the weekend of June 10-11. The FreeBSD Foundation sponsored several users who have summarized their experiences.

2016 heralded my return to BSDCan after a 4 year hiatus. In part, I was inspired to return this year, after I took some holidays in France back in February. I had the distinct pleasure, that weekend, to have supper with just about all the Paris based FreeBSD committers. Plus, I got to meet my first ever mentee, jadawin@, who made a special trip to come visit. Keeping company with these great people for the night reminded me of the great camaraderie I had experienced at the conferences in years gone by, and I wanted to try experience that again. So, I showed up on campus, and everything felt familiar. This was the first good sign! From there, a few familiar faces were revealed, and before too long, it was almost old homecoming for me!

Thomas M. Abthorpe:

<https://www.freebsdoundation.org/blog/2016-bsdcan-trip-report-thomas-m-abthorpe/>

Trent Thompson:

<https://www.freebsdoundation.org/blog/2016-bsdcan-trip-report-trent-thompson/>

Li-Wen Hsu: <https://www.freebsdoundation.org/blog/2016-bsdcan-trip-report-li-wen-hsu/>

Ruslan Bukin: <https://www.freebsdoundation.org/blog/2016-bsdcan-trip-report-ruslan-bukin/>

<https://www.freebsdnews.com/2016/07/25/2016-bsdcan-trip-report/>

Red Hat, Microsoft and Codenvy Push DevOps with New Language Protocol

Red Hat, Microsoft and Codenvy partnered recently to knock down another barrier to open collaboration through the launch of a new open source project called Language Server Protocol. Here's what it means for the channel.

The three companies describe the protocol, which they announced in June, as "an open source project that defines a JSON-based data exchange protocol for language servers, which can provide programming language services like Find By Symbol or Refactoring consistently across different code editors. This protocol is accessible over standard I/O, allowing both locally installed and remotely hosted editors to access these features, running inside a language server."

What that means in non-technical terms is that the protocol provides a common, open standard for allowing developers to use any type of programming language with any type of programming app.

With the Language Server Protocol, "developers can gain access to intelligence for any language within their favorite tools," according to Jewell.

What It Means for the Channel: Advancing DevOps and Open Source

To be sure, the Language Server Protocol is something that only programmers are likely to appreciate fully.

Yet the project is also important from a broader channel perspective, for two reasons.

First, it's interesting as the latest partnership between erstwhile enemies Microsoft and Red Hat. While it's no longer news that Microsoft wants to cooperate with the open source community, it's remarkable that Redmond is now going so far as to help found an open source project whose goal is to erase platform lock-in for programming. In the past, platform lock-in constituted the crux of Microsoft's business strategy, but those days are long past.

Second, this news is evidence of how DevOps practices are revolutionizing the channel. The Language Server Protocol is the latest in a series of DevOps tools designed to make app development and delivery more modular and platform-agnostic, while freeing programmers to use whichever toolset they decide is best for the job at hand. Vendors who want to prepare for the future need to adopt the same mindset.

<http://thevarguy.com/open-source-application-software-companies/red-hat-microsoft-and-codenvy-push-devops-new-language-pr>

Top Apple Open Source Projects You Must Know

Today's industry player is the increasingly controversial Apple. Although Apple doesn't advertise it, Apple has a long-time strong relationship with open source communities. Apple contributes to many open source projects as they incorporate them into iOS and the newly branded macOS, not to mention the pillars of the Apple operating systems being a mix-up of FreeBSD, the Mach Kernel, and the Darwin Kernel, plus much more open source software like the GNU Utils.

The benefits of opening up code come two-fold and tend to feedback into itself. First, the public benefits when the code is opened simply by having access. Then, the author benefits because the public can make recommendations, and possibly even changes. When the codebase becomes better as the result of the dialogue generated by the public forum around the code, it draws more attention. This is how the humble Linux kernel started and came to dominate the world of operating systems.

So, let's take a look at the list of top Apple open source projects:

Swift



In 2014, Apple shocked the world with the announcement of its Swift programming language. Swift is a modern programming language with loads of features. It has seen unparalleled adoption rates and boasts quite out-of-the-box library considering it can leverage both C and Objective-C libraries and frameworks. Apple surprised the world, yet again, when they decided to open source their new language. Since then, Swift has gained popularity on Apple

and Linux platforms.

WebKit



Initially released in 1998 as KHTML, and part of the KDE project, WebKit has been around for quite some time. WebKit is the rendering engine that powers Safari, both desktop and mobile, as well as Google Chrome, desktop and mobile. WebKit has extensive standards support while maintaining performance, which is key with the sheer amount of media in modern websites. WebKit is a powerful piece of technology that continues to deliver.

ResearchKit and CareKit



These are two frameworks that are nothing but good intentions. ResearchKit is a framework that will allow medical professionals to develop applications that can accurately track and measure illness and disease to an unprecedented degree, and combining with CareKit, it puts power in the hands of the patients

themselves. Patients can easily supply their doctors with day-to-day updates pertaining to the progression or recession of medical conditions. This benefits both the patients as well as the medical research community. These two open source frameworks can potentially revolutionize medicine.

It's easy to see that Apple takes open source seriously. They're major contributors, and not just to the projects they lead. Be sure to check out the links provided to see where else Apple contributes as well as the contributions of other companies.

<http://fossbytes.com/top-apple-open-source-projects-must-know/>

US.gov to open-source made-to-order software, allow contributions

Three-year code pilot to cut costs, lock-in

United States Chief Information Officer Tony Scott and Chief Acquisition Officer Anne E Rung have issued a joint memo decreeing that henceforth all government agencies need to consider open-sourcing any bespoke software they commission.

The memo (PDF), issued on Monday, notes some code-sharing across government agencies but says it is not done "in a consistent manner".

"In some cases, agencies may even have difficulty establishing that the software was produced in the performance of a Federal Government contract," the memo continues, which can lead to "duplicative acquisitions for substantially similar code and an inefficient use of taxpayer dollars".

The policy therefore implements a three-year pilot during which US government agencies will be required to open source a fifth of their bespoke code. Security agencies are exempt from the policy.

The policy also calls for any bespoke development effort to “acquire and enforce rights sufficient to enable Government-wide reuse of custom-developed code.” There's also a requirement to keep an up-to-date inventory of code and to lodge open source code at code.gov.

Elsewhere, the policy suggests that when sharing code, agencies should engage with existing communities whenever possible, rather than trying to create their own. Which sounds like a shout-out to whoever provisions storage at GitHub, if nothing else. There's even a section 5.2.F in which agencies are encouraged to ready themselves for code contributions from third parties within and without government, creating the potential for citizen coders to help build government apps.

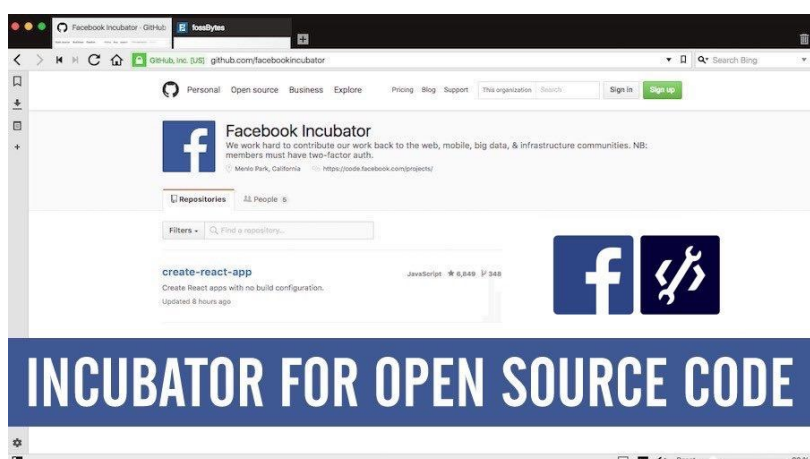
The memo also insists that whenever agencies need new software they must consider “whether to use an existing Federal software solution or to acquire or develop a new software solution.” Agencies must also consider whether it is possible to get what they need by mixing government and commercial code.

Similar policies have sprouted around the world, often accompanied by the concept of a government app store, so the US isn't out on its own here. The sheer size of the US government, however, means the concept has just leveled up.

The memo's authors hope agencies do, too: the 20 per cent target is suggested as a minimum and “agencies are strongly encouraged to release as much custom-developed code as possible to further the Federal Government's commitment to transparency, participation, and collaboration.”

http://www.theregister.co.uk/2016/08/09/us_government_to_open_source_bespoke_code_and_a_low_contributions/

“Incubator” On GitHub — Facebook’s Open Source Gift To Programmers



Introducing a new way of thinking (that should be adopted by other tech giants) while working with open source projects, Facebook has launched Incubator on GitHub. The social network aims to release its internal open source projects via this central channel and observe their adoption in the open source community. If a project does well and gains popularity, it'll graduate to its own repo.

GitHub is the most popular web-based GIT repository hosting service. It lets the developers host their code and collaborate on open source projects. But, a big company's big bucket of new and unstructured code could easily seem invisible (and get lost) to a novice developer.

To solve this problem, the social media giant Facebook has decided to give a bit of structure to its open source code with its new Incubator hub on GitHub.

It's basically a new process of releasing new open source projects to the developer community. By using Incubator as a proving ground, Facebook aims to make sure that these projects are adopted well.

Last week, Facebook launched a new project named Create React App to help React developers get started with new projects easily. Create React App was the first project to be inducted into the Facebook Inductor on GitHub.

With this gateway, Facebook plans to push more open source projects and see how developers react to them. These projects will be the ones that are used by the social network internally to improve its services.

If a project manages to gain enough traction from the open source community, it will graduate into its own standalone repository.

Notably, Facebook Incubator is just for Facebook's own projects. But everyone can take advantage of open projects being pushed here. It will not only provide repos, but also a new way of thinking while working with open source projects.

So, what do you think about the new Facebook Incubator on GitHub? Share your views in the comments section below.

<http://fossbytes.com/incubator-github-facebooks-gift-open-source-developers/>

FreeBSD Mid-Year Fundraising Campaign Kick-Off

I'm pleased to announce the kickoff of our mid-year fundraising campaign! We are more than half-way through the year, but we've only raised \$265,000 towards our goal of raising \$1,250,000. We are reaching out to you, the FreeBSD community, to help us promote our work and to make a donation so we can continue supporting FreeBSD.

Last year, we raised \$656,594 and spent \$1,093,204. After reviewing where we spent our money to determine our budget for 2016, we decided to continue investing in areas of the Project that are important for continuing its path of growth and innovation.

Our purpose is to help enable the success of FreeBSD as a platform for product development, education, research, and end-users. We're doing this by providing assistance to the Project in critical areas and roles, outreach and advocacy, facilitating community collaboration and engagement, and helping to keep FreeBSD secure, stable, and reliable.

Your donations will directly support FreeBSD by helping us:

- Provide outreach and advocacy for FreeBSD, which includes sponsoring many BSD and non-BSD conferences; sending FreeBSD contributors to these conferences; improving the new user experience; supporting work on creating curriculum to be taught in schools and universities; publishing the high-quality FreeBSD focused magazine, The FreeBSD Journal; and providing more informational and training material.
- Make OS improvements, including employing our technical staff to maintain and improve critical kernel subsystems, add features and functionality, and fix problems. This also includes funding larger projects, like the arm64 port and toolchain work, to make sure FreeBSD remains a viable solution for new platforms and technologies.
- Support the security team by improving processes and policies, and by bolstering the operational capacity of the team.
- Organize and run vendor summits and visit companies to help facilitate collaboration between commercial users and the Project, to help get changes pushed into the FreeBSD source tree, and to create a bigger and healthier ecosystem.
- Improve developer infrastructure to help modernize the tools and make contributions more efficient.
- Provide full-time release engineering support, resulting in on-time and reliable releases.

Your passion is what helps drive us to do the work that we do.

Please consider making a donation today, talking to your company to make a donation, and spreading the word about our fundraising campaign.

Thank you for your support – we can't do this without you!

Source: FreeBSD Foundation

https://bsdmag.org/donate_freebsd/

Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES

1.925.240.6652

\$39.95

FreeBSD 9.1 Jewel Case CD Set
or FreeBSD 9.1 DVD

\$29.95

PC-BSD 9.1 DVD

\$49.95

The PC-BSD 9.0 Users Handbook
PC-BSD 9.1 DVD

\$99.95

The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.1 CD or DVD set
FreeBSD Toolkit DVD



Stylish Dress Attire
Look Your Professional Best



Comfy Apparel
Stay Warm in Zip Ups & Pullovers

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.1 Jewel Case CD/DVD \$39.95

CD Set Contains:

- Disc 1** Installation Boot LiveCD (i386)
- Disc 2** Essential Packages Xorg (i386)
- Disc 3** Essential Packages, GNOME2 (i386)
- Disc 4** Essential Packages (i386)

FreeBSD 9.0 CD \$39.95

FreeBSD 9.0 DVD \$39.95

FreeBSD Subscriptions

Save time and \$\$\$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1 \$29.95

FreeBSD Subscription, start with DVD 9.1 \$29.95

FreeBSD Subscription, start with CD 9.0 \$29.95

FreeBSD Subscription, start with DVD 9.0 \$29.95

PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD \$29.95

PC-BSD Subscription \$19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide) \$39.95

The FreeBSD Handbook, Volume 2 (Admin Guide) \$39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes) \$59.95

The FreeBSD Handbook, Both Volumes & FreeBSD 9.1 \$79.95

PC-BSD 9.0 Users Handbook \$24.95

BSD Magazine \$11.99

The FreeBSD Toolkit DVD \$39.95

FreeBSD Mousepad \$10.00

FreeBSD & PCBSD Caps \$20.00

BSD Daemon Horns \$2.00



Bundle Specials!
Save \$\$\$

Just Plain Fun
Mousepads & Novelty Horns



BSD Magazine
Available Monthly



For even MORE items
visit our website today!

www.FreeBSDMall.com

MINIX 3 Free, Open-Source, Operating System, Highly Reliable, Flexible and Secure

by Mauro Risonho de Paula Assumpção

MINIX 3 was publicly announced on 24 October 2005 by Andrew Tanenbaum during his keynote speech at the ACM Symposium Operating Systems Principles conference. Although it still serves as an example for the new edition of Tanenbaum and Woodhull's textbook, it is comprehensively redesigned to be "usable as a serious system on resource-limited and embedded computers and for applications requiring high reliability."

Reliability in MINIX 3

One of the main goals of MINIX 3 is reliability. Below, some of the more important principles that enhance MINIX 3's reliability are discussed.

Reduced kernel size

Monolithic operating systems, such as Linux and FreeBSD, and hybrids like Windows, have millions of lines of kernel code. In contrast, MINIX 3 has about 6,000 lines of executable kernel code, which can make problems easier to find in the code.

Cage the bugs

In monolithic kernels, device drivers reside in the kernel itself. This means that when a new peripheral is installed, unknown, untrusted code is inserted in the kernel. A single bad line of code in a driver can bring down the system.

In MINIX 3, each device driver is a separate user-mode process. Drivers cannot execute privileged instructions, change the page tables, perform arbitrary input/output (I/O), or write to absolute memory. They have to make kernel calls for these services and the kernel checks each call for authority.

Limit drivers' memory access

In monolithic kernels, a driver can write to any word of memory and thus accidentally trash user programs.

In MINIX 3, when a user expects data from, for example, the file system, it builds a descriptor telling who has access and at what addresses. It then passes an index of this descriptor to the file system, which may pass it to a driver. The file system or driver then asks the kernel to write via the descriptor, making it impossible for them to write to addresses outside the buffer.

Survive bad pointers

Dereferencing a bad pointer within a driver will crash the driver process, but will have no effect on the system as a whole. The reincarnation server will restart the crashed driver automatically. For some drivers (e.g., disk and network), recovery is transparent to user processes. For others (e.g., audio and printer), the user may notice. In monolithic kernels, dereferencing a bad pointer in a driver normally leads to a system crash.

Tame infinite loops

If a driver gets into an infinite loop, the scheduler will gradually lower its priority until it becomes idle. Eventually, the reincarnation server will see that it is not responding to status requests, so it will kill and restart the looping driver. In a monolithic kernel, a looping driver could hang the system.

Limit damage from buffer overflows

MINIX 3 uses fixed-length messages for internal communication, which eliminates certain buffer overflows and buffer management problems. Also, many exploits work by overrunning a buffer to trick the program into returning from a function call using an overwritten stack return address pointing into attacker controlled memory, usually the overrun buffer itself. In MINIX 3, this attack is mitigated because instruction and data space are split and only code in (read-only) instruction space can be executed, commonly known as Data Execution Prevention. However, attacks that rely on running legitimately executable memory in a malicious way (return-to-libc, Return-oriented programming) are not prevented by this mitigation.

Restrict access to kernel functions

Device drivers obtain kernel services (such as copying data to users' address spaces) by making kernel calls. The MINIX 3 kernel has a bit map for each driver specifying which calls it is authorized to make. In monolithic kernels, every driver can call every kernel function, authorized or not.

Restrict access to I/O ports

The kernel also maintains a table telling which I/O ports each driver may access. As a result, a driver can only touch its own I/O ports. In monolithic kernels, a buggy driver can access I/O ports belonging to another device.

Restrict communication with OS components

Not every driver and server needs to communicate with every other driver and server. Accordingly, a per-process bit map determines which destinations each process may send to.

Reincarnate dead or sick drivers

A special process, called the reincarnation server, periodically pings each device driver. If the driver dies or fails to respond correctly to pings, the reincarnation server automatically replaces it with a fresh copy. The detection and replacement of non-functioning drivers is automatic, without any user action required. This feature does not work for disk drivers at present, but in the next release the system will be able to recover even disk drivers, which will be shadowed in random-access memory (RAM). Driver recovery does not affect running processes.

Integrate interrupts and messages

When an interrupt occurs, it is converted at a low level to a notification sent to the appropriate driver. If the driver is waiting for a message, it gets the interrupt immediately; otherwise, it gets the notification the next time it does a RECEIVE to get a message. This scheme eliminates nested interrupts and makes driver programming easier.

General Characteristics

- POSIX-compliant operating system with a NetBSD userland
- Open source, with a BSD license
- Runs on x86 PCs and well as x86 virtual machines (VMware, etc.)
- Runs on ARM Cortex A8 (e.g., BeagleBoard XM, Beaglebones)
- Networking with TCP/IP

- Virtual memory
- Virtual file system
- Unified block cache shared by virtual memory and file systems
- Dynamic linking
- Small memory footprint (kernel is 600 kB; full is 25)

MINIX-specific Features

- Tiny microkernel that runs in kernel mode
- Most of the operating system runs in user-mode protected processes
- Each device driver is a separate user-mode process
- Reincarnation server can reload failed drivers

Reliability Features

- Reduced kernel size
- Bugs are caged
- Drivers' memory access is limited
- Bad pointer references are not always fatal
- Infinite loops are not always fatal
- Buffer overruns are not always fatal
- Access to kernel function calls is restricted
- Access to I/O ports is restricted
- Communication with components is restricted
- Dead or sick drivers can be reincarnated
- Interrupts and messages are integrated
- Languages and Compilers
- Languages: C, C++, clisp, mawk, Perl, Python, tcl, etc.

- Compilers: gcc and clang/LLVM
- Native compilation (self hosting) on x86
- Cross compilation for x86 and ARM

Packages

- Shells (e.g., bash, mksh, mudsh, pdksh, zsh)
- Editors (e.g., elvis, joe, jove, pico, uemacs, vim)
- Games (e.g., crafty, exchess, ioquake)
- Mail (e.g., fetchmail, getmail, mutt, thunderbird)
- Over 4000 other NetBSD packages

Installing and Running MINIX 3 on VirtualBox

This page describes the process of installing MINIX 3 on VirtualBox.

Getting MINIX

Download the CD-ROM installer image:

- minix_R3.3.0-588a35b.iso

Version	Medium	Image	Image Torrent	md5sum
3.3.0 (stable release)	CD-ROM	288 MB	torrent	3234ffcebf2a28069cf3def41c95dec
3.2.1 (previous)	CD-ROM	256 MB	torrent	4c91ba7822cfa441d27755a7e7c4711d

- Decompress the downloaded file to get an .iso file and the Installation guide (same as above).
- Burn this bootable CD-image file to a CD-ROM.
- Reboot the computer with the CD-ROM device and follow the instructions in the installation guide (same as above).

Preliminaries

First of all, you'll need to install VirtualBox(<https://www.virtualbox.org/>). VirtualBox binaries can be downloaded from their webpage. If you're running a Linux distribution, you can install VirtualBox via the package manager.

Virtual Machine Setup

Before you install MINIX 3, you will need to create a new virtual machine configuration. The VM configuration specifies the parameters of your Virtual machine, e.g., how much memory you want the VM to use, how big you want the virtual hard disk to be, etc. Please see Hardware Requirements (<http://wiki.minix3.org/doku.php?id=usersguide:hardwarerequirements>) for guidelines.

In the main screen of VirtualBox, click the big New button.

1. At the Name and operating system screen, for Name write MINIX3 (anything will work). For Type and Version select Other.
2. At the Memory size screen, select the amount of memory for this Virtual Machine.
3. At the Hard Drive screen, set the size and properties of the Virtual Hard Disk. It is okay to either leave those options at their defaults or change them.
4. Pressing Create will create the Disk Image and the Virtual Machine that we will run.
5. Now select MINIX3 in the list on the left.
6. Click the Settings button on the main screen of VirtualBox. Then click on System in the list on the right and tick the Hardware Clock in UTC Time checkbox.
7. Click OK, and you are now ready to install MINIX 3!

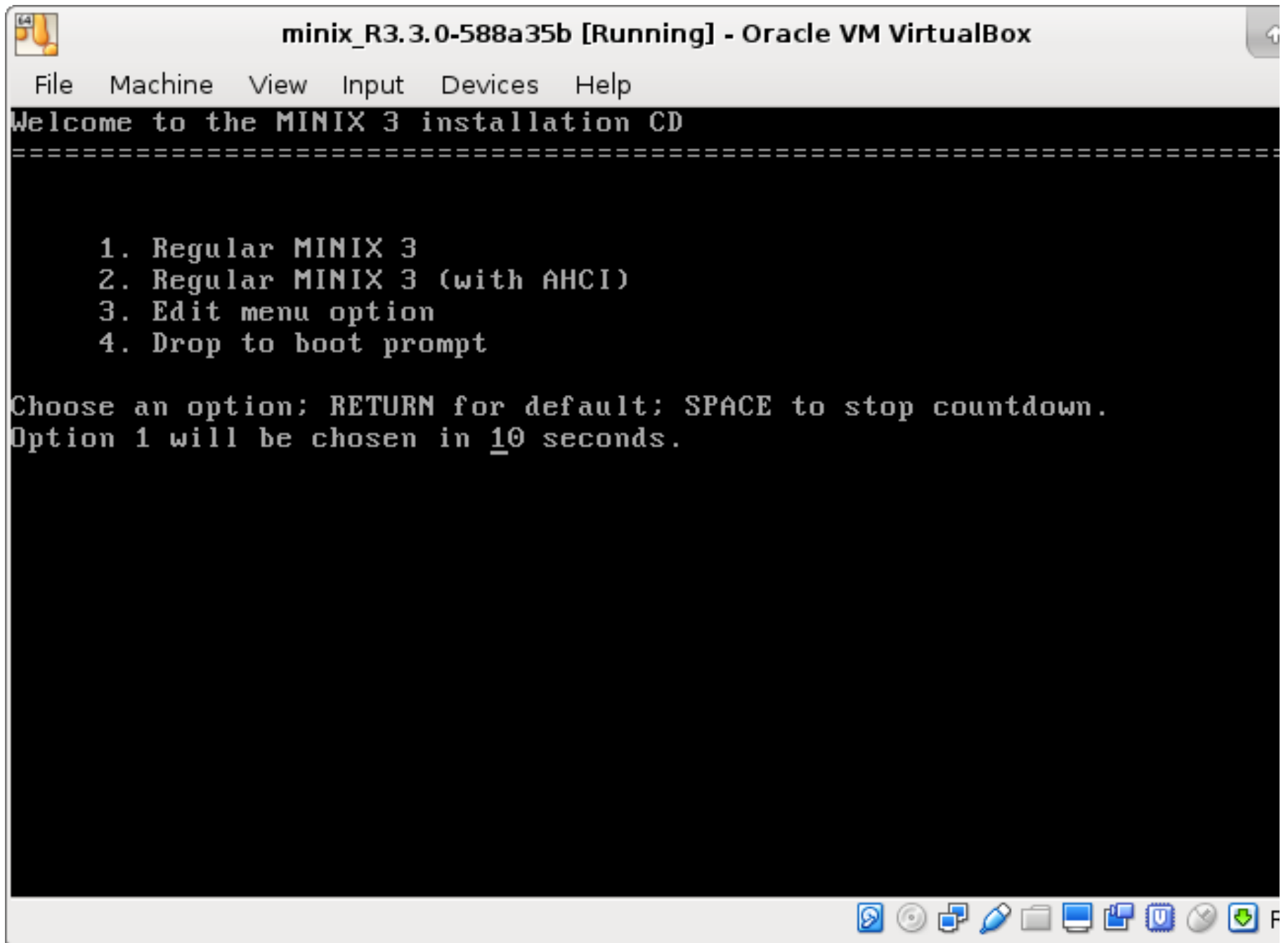
Installation

Assuming you have downloaded and decompressed a MINIX 3 ISO image from the download page (<http://www.minix3.org/download>), you can mount the ISO file:

1. Select MINIX3 in the list on the left.
2. Click Start.
3. You will be asked to select a start-up disk. Browse to and select the .iso MINIX image you downloaded earlier and press Open.

Installing

These steps correspond to the steps on the screen.

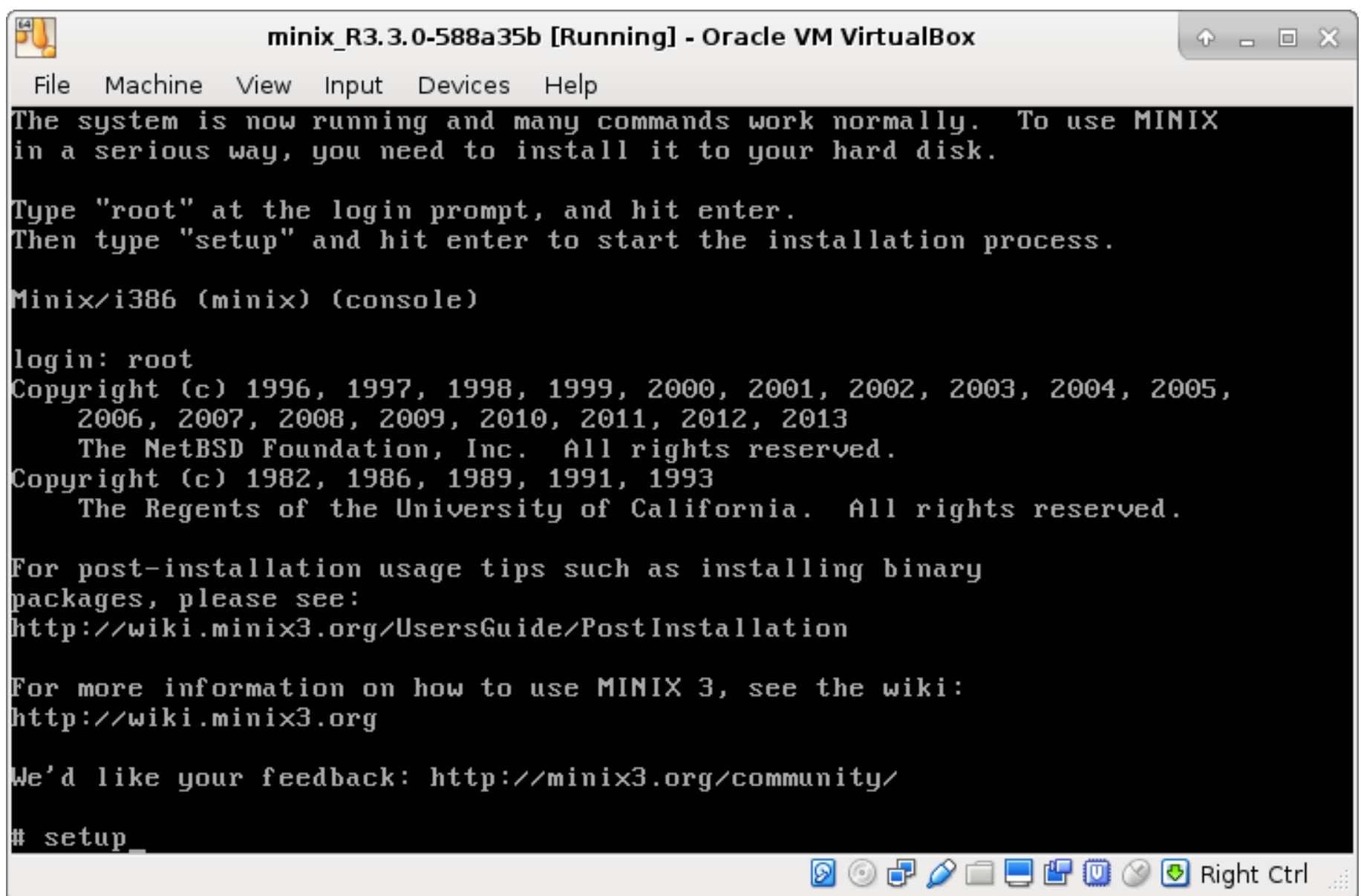


Running the Setup script

When the login prompt appears, login as root. Press Enter when prompted for a password.

To start the installation of MINIX on the hard disk, type

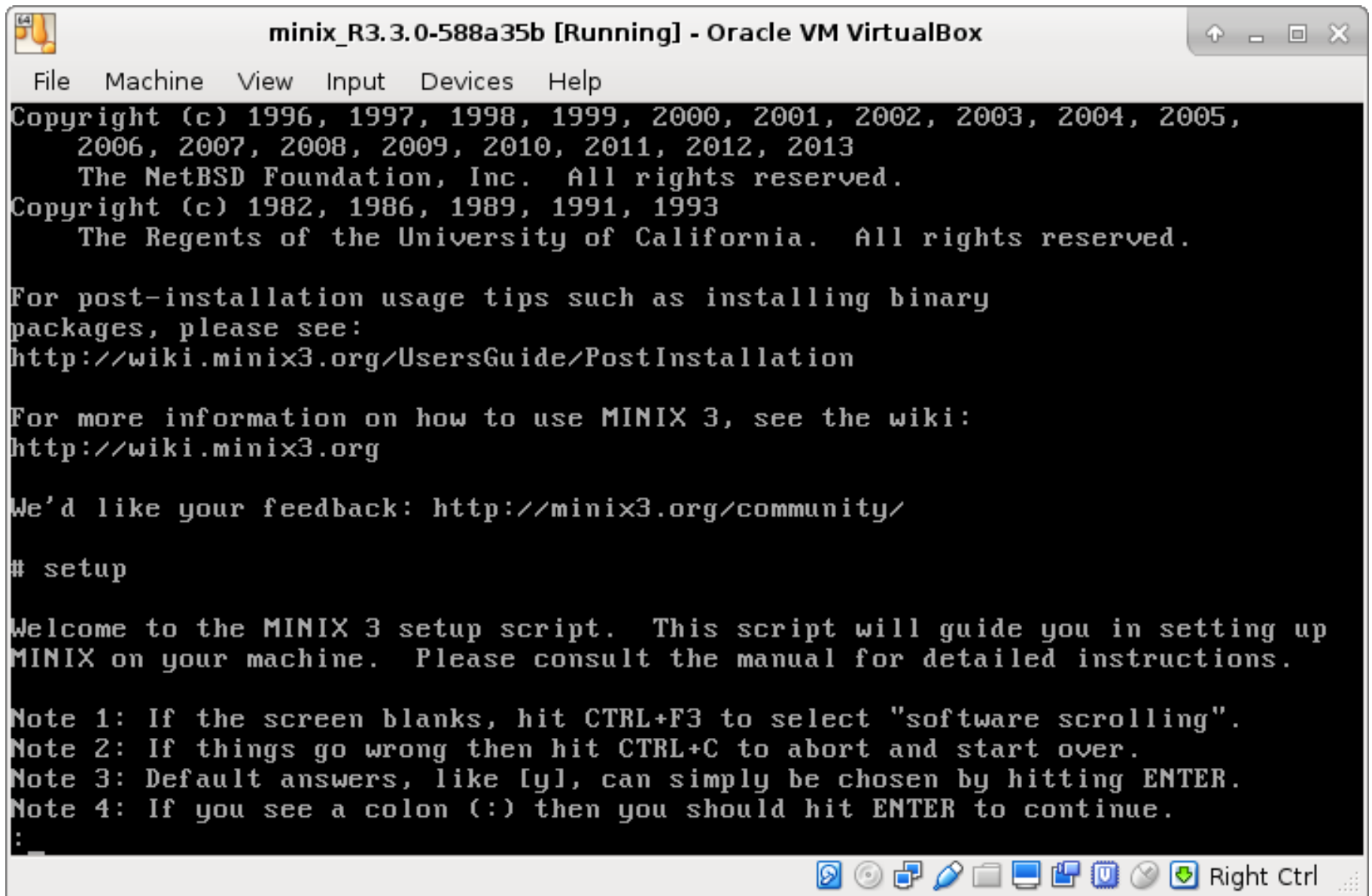
```
setup
```

```
minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
The system is now running and many commands work normally. To use MINIX
in a serious way, you need to install it to your hard disk.
Type "root" at the login prompt, and hit enter.
Then type "setup" and hit enter to start the installation process.
Minix/i386 (minix) (console)
login: root
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013
The NetBSD Foundation, Inc. All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
The Regents of the University of California. All rights reserved.
For post-installation usage tips such as installing binary
packages, please see:
http://wiki.minix3.org/UsersGuide/PostInstallation
For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org
We'd like your feedback: http://minix3.org/community/
# setup_
```

After this and all other commands, be sure to press ENTER (RETURN). When the installation script ends, you should see a screen with a colon prompt; hit ENTER to continue.

If the screen suddenly goes blank, press CTRL-F3 to select software scrolling (should only be needed on very old computers). Note that CTRL-key means depress the CTRL key and while holding it down, press "key."



```
minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005,
2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013
The NetBSD Foundation, Inc. All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
The Regents of the University of California. All rights reserved.

For post-installation usage tips such as installing binary
packages, please see:
http://wiki.minix3.org/UsersGuide/PostInstallation

For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org

We'd like your feedback: http://minix3.org/community/

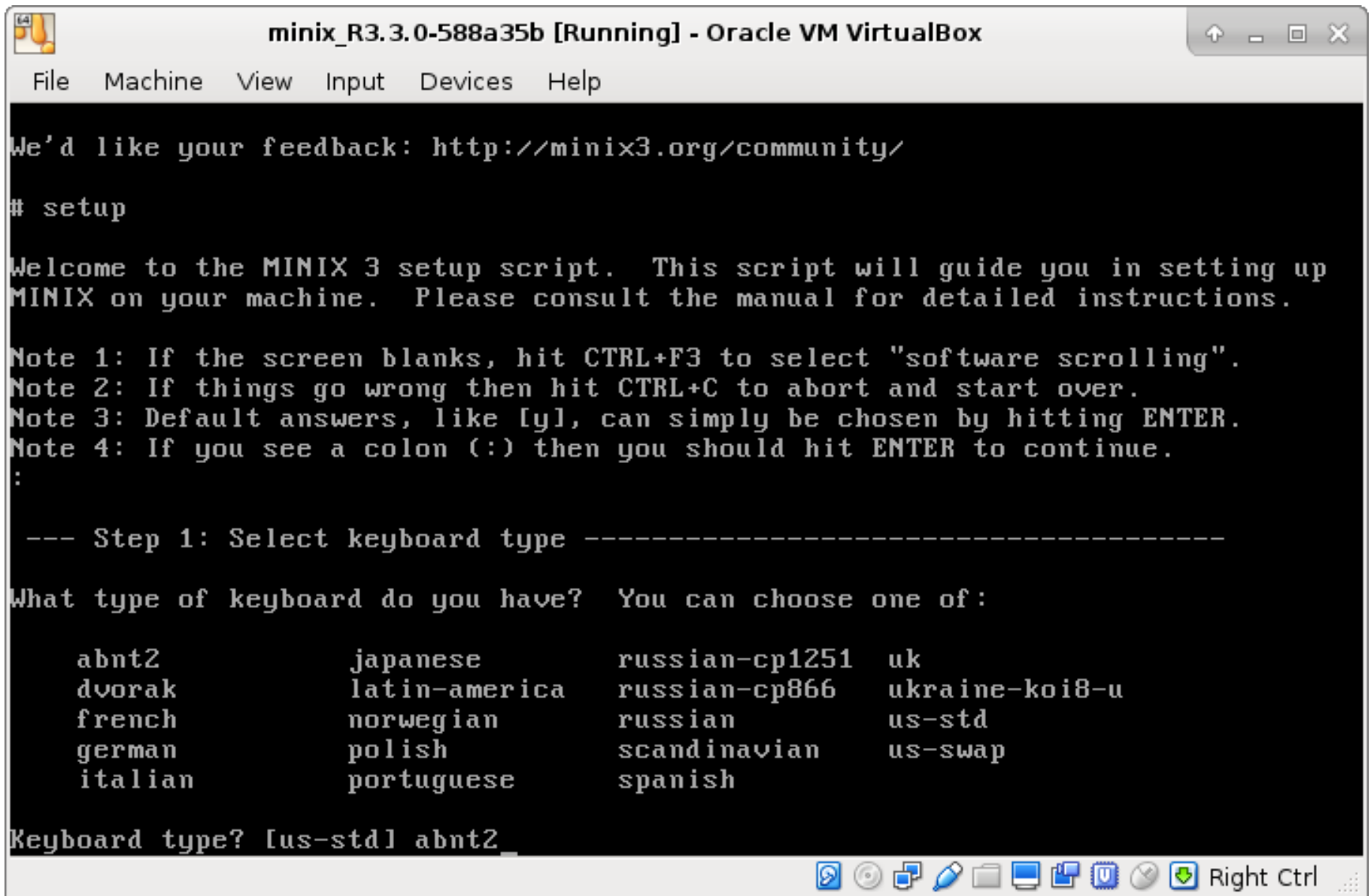
# setup

Welcome to the MINIX 3 setup script. This script will guide you in setting up
MINIX on your machine. Please consult the manual for detailed instructions.

Note 1: If the screen blanks, hit CTRL+F3 to select "software scrolling".
Note 2: If things go wrong then hit CTRL+C to abort and start over.
Note 3: Default answers, like [y], can simply be chosen by hitting ENTER.
Note 4: If you see a colon (:) then you should hit ENTER to continue.
:
_
```

Select keyboard type

When you are asked to select your national keyboard, do so. This and other steps have a default choice, in square brackets. If you agree with it, just hit ENTER. In most steps, the default is generally a good choice for beginners. The us-swap keyboard interchanges the CAPS LOCK and CTRL keys, as is conventional on UNIX systems.



```
minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

We'd like your feedback: http://minix3.org/community/

# setup

Welcome to the MINIX 3 setup script. This script will guide you in setting up
MINIX on your machine. Please consult the manual for detailed instructions.

Note 1: If the screen blanks, hit CTRL+F3 to select "software scrolling".
Note 2: If things go wrong then hit CTRL+C to abort and start over.
Note 3: Default answers, like [y], can simply be chosen by hitting ENTER.
Note 4: If you see a colon (:) then you should hit ENTER to continue.
:

--- Step 1: Select keyboard type -----

What type of keyboard do you have? You can choose one of:

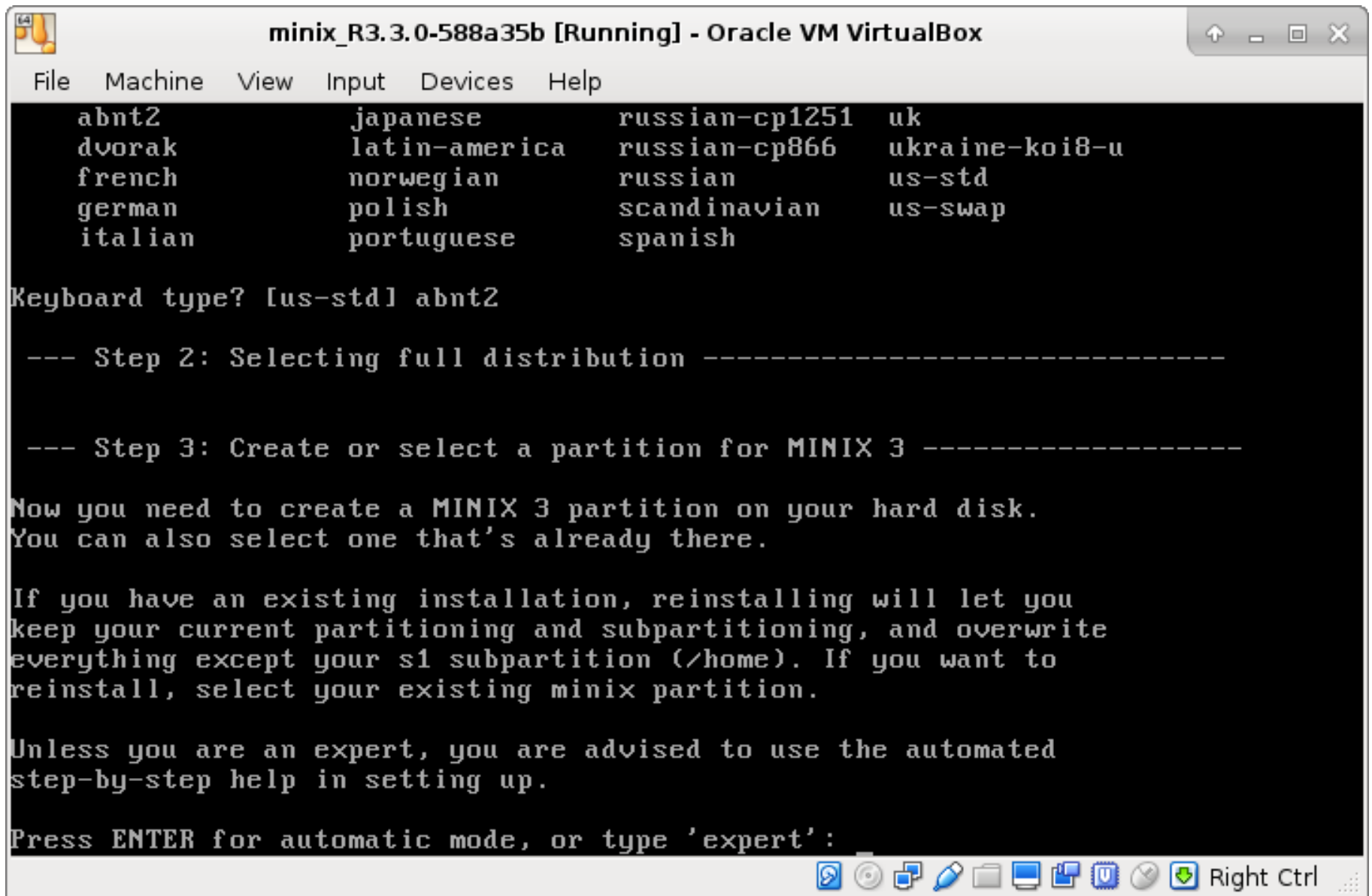
    abnt2      japanese    russian-cp1251  uk
    dvorak     latin-america  russian-cp866  ukraine-koi8-u
    french     norwegian     russian         us-std
    german     polish        scandinavian   us-swap
    italian    portuguese    spanish

Keyboard type? [us-std] abnt2_
```

In this case, choose the keyboard but is similar to the language of their country. As this article will be "ABNT2", Brazil

Create or select a partition for MINIX

You will first be asked if you are an expert in MINIX disk partitioning. If so, you will be placed in the part of the program to give you full power to edit the Master Boot Record (and enough rope to hang yourself). If you are not an expert, press ENTER for the default action, which is an automated step-by-step guide to formatting a disk partition for MINIX.



Select a disk

An IDE controller may have up to four disks. The setup script will now look for each one. Just ignore any error messages. When the drives are listed, select one and confirm your choice.

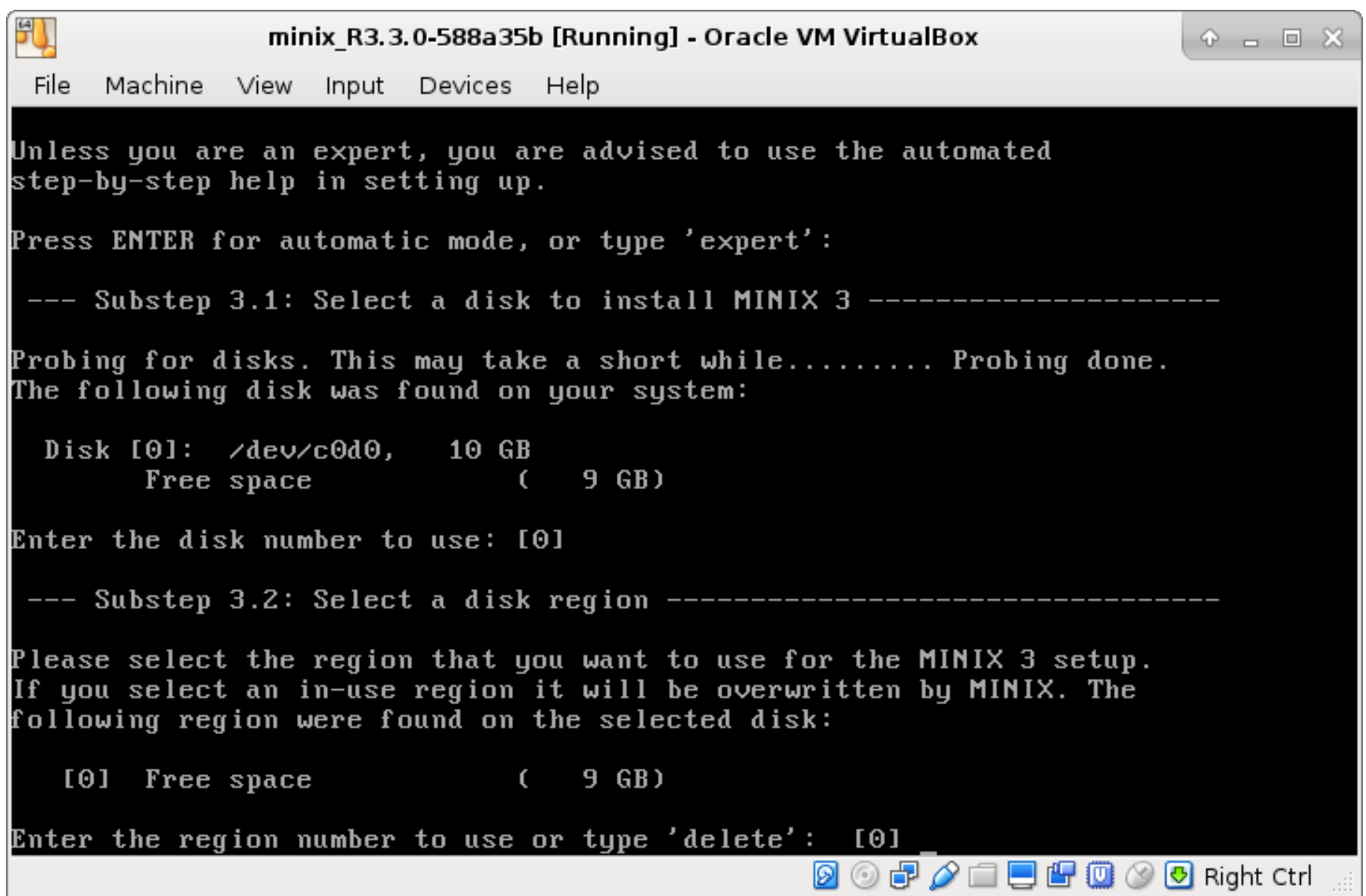
Select a disk region

Now choose a region to install MINIX into. You have three choices:

1. Select a free region
2. Select a partition to overwrite
3. Delete a partition to free up space and merge with adjacent free space

For choices (1) and (2), type the region number. For (3) type:

```
delete
```



```
minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Unless you are an expert, you are advised to use the automated
step-by-step help in setting up.

Press ENTER for automatic mode, or type 'expert':

--- Substep 3.1: Select a disk to install MINIX 3 -----

Probing for disks. This may take a short while..... Probing done.
The following disk was found on your system:

Disk [0]: /dev/c0d0, 10 GB
Free space ( 9 GB)

Enter the disk number to use: [0]

--- Substep 3.2: Select a disk region -----

Please select the region that you want to use for the MINIX 3 setup.
If you select an in-use region it will be overwritten by MINIX. The
following region were found on the selected disk:

[0] Free space ( 9 GB)

Enter the region number to use or type 'delete': [0]
```

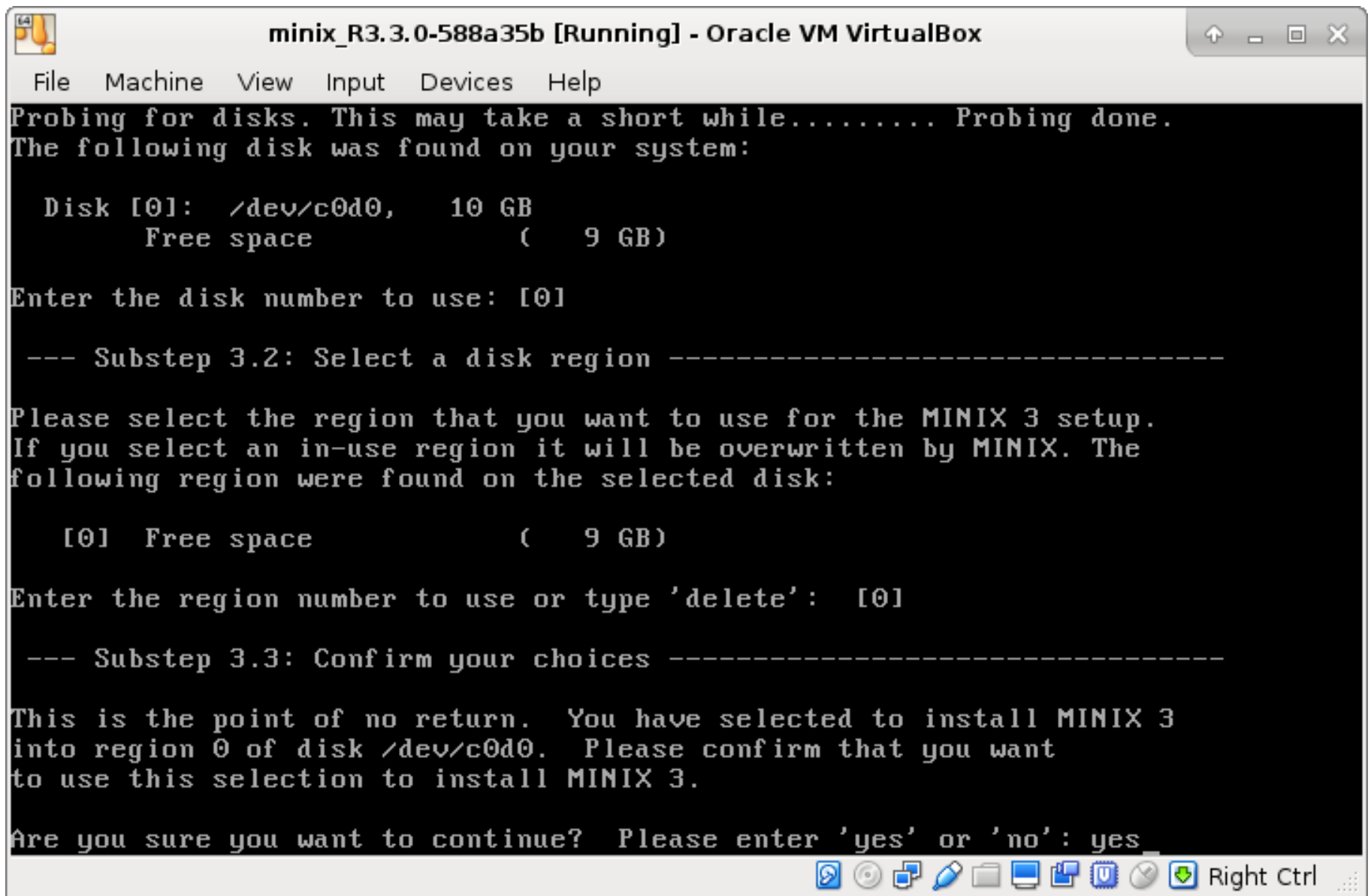
Then give the region number when asked. This region will be overwritten and its previous contents lost forever.

Confirm your choices

You have now reached the point of no return. You will be asked if you want to continue. If you do, the data in the selected region will be lost forever. If you are sure, type:

```
yes
```

and then press ENTER. To exit the setup script without changing the partition table, hit `CTRL-C`.



Reinstall choice

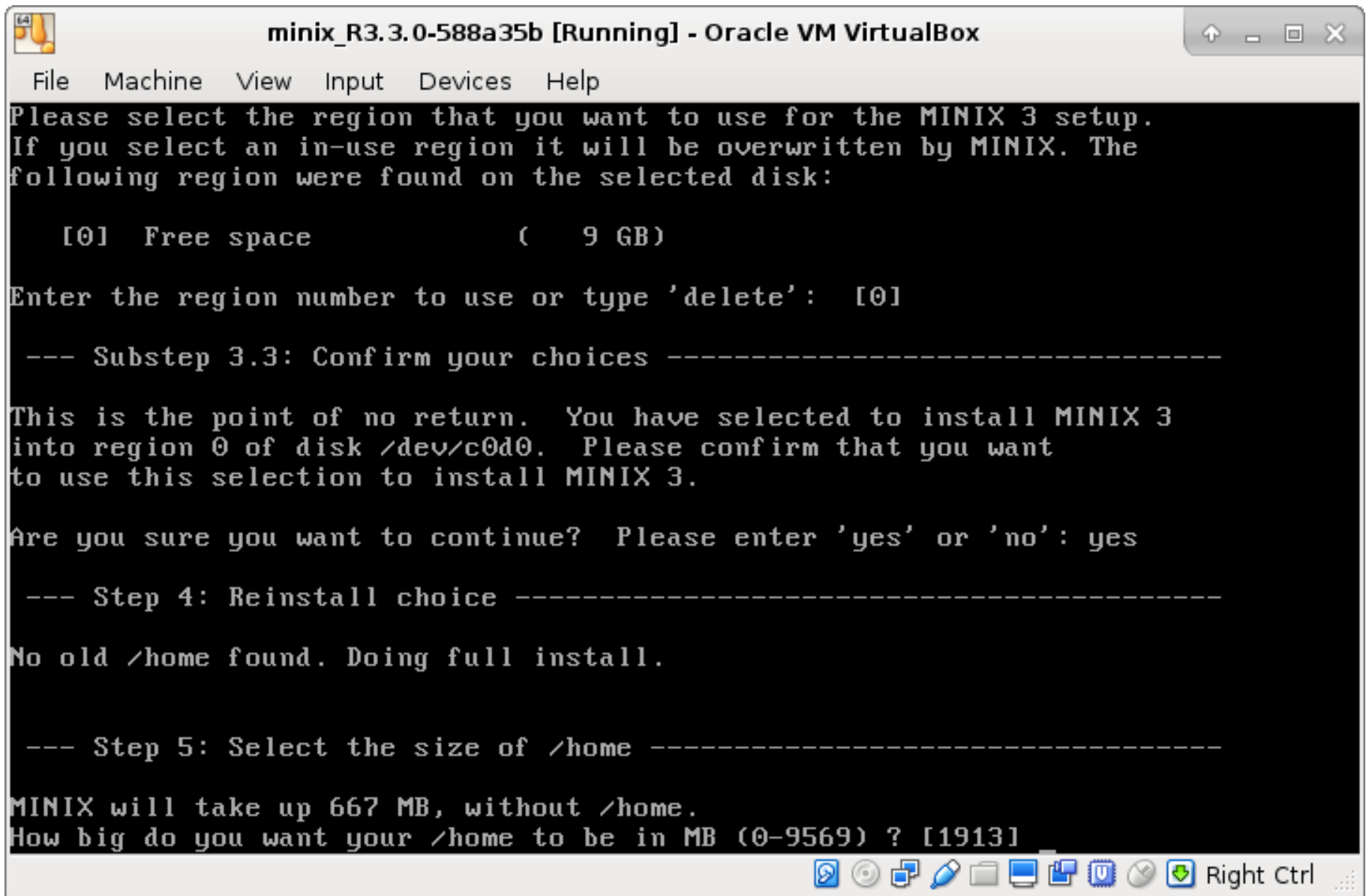
If you chose an existing MINIX partition, in this step you will be offered a choice between a Full install, which erases everything in the partition, and a Reinstall, which does not affect your existing /home partition.

This design means that you can put your personal files on /home and reinstall a newer version of MINIX when it is available without losing your personal files.

Select the size of /home

The selected partition will be divided into three subpartitions: root, /usr, and /home.

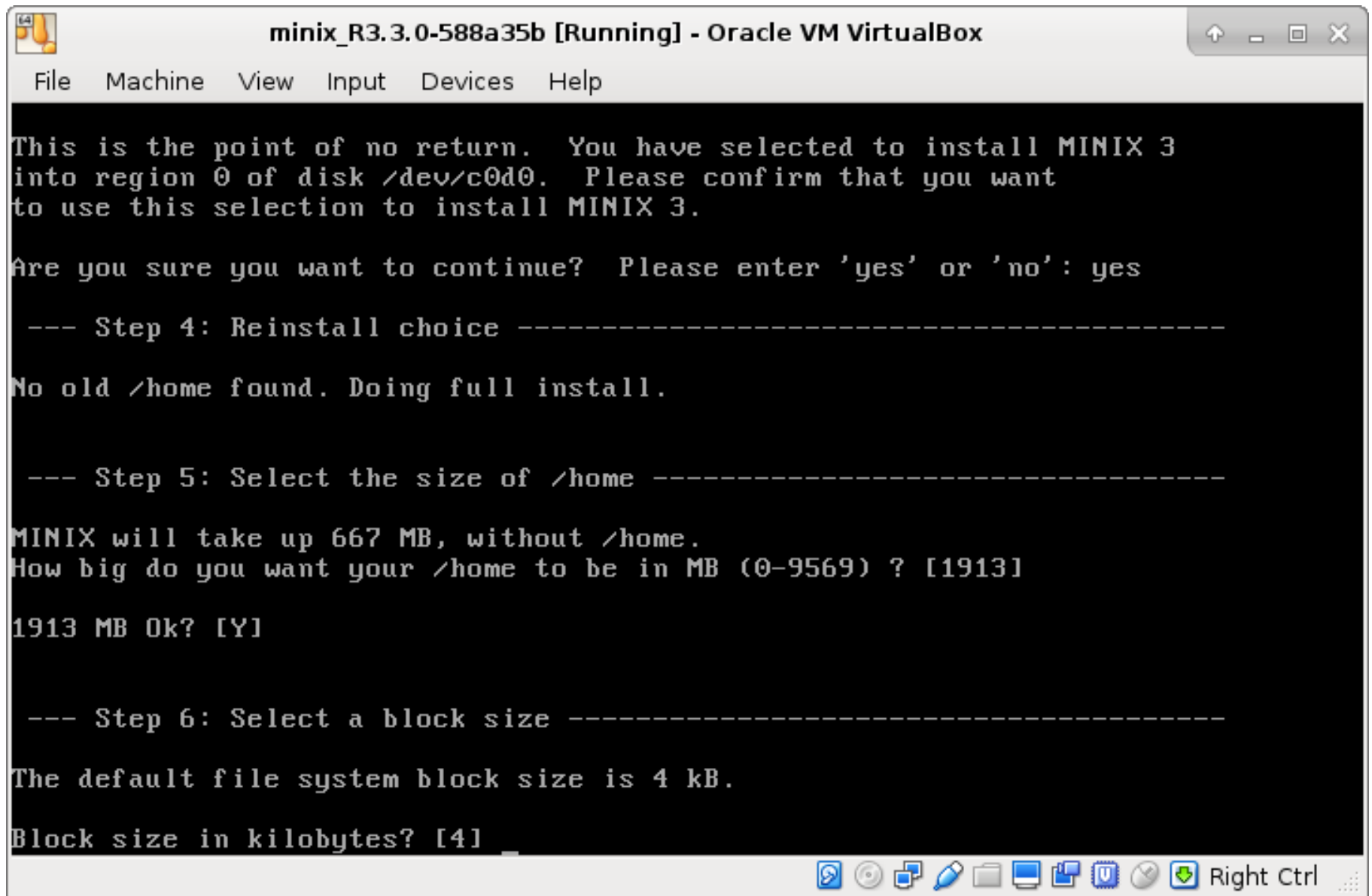
- /home, will contain only your own personal files. Specify how much of the partition should be set aside for your files. You will be asked to confirm your choice.



- `/usr` contains most of the software, as well as all the optional packages. It is advised to give it several gigabytes if possible. Its size is computed as the remaining space on the MINIX partition, so the bigger `/home` is, the smaller `/usr` is.

Select a block size

Disk block sizes of 1-KB, 2-KB, 4-KB, and 8-KB are supported, but to use a size larger than 4-KB you have to change a constant and recompile the system. Use the default (4 KB) here.



The screenshot shows a terminal window titled "minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox". The terminal displays the following text:

```
This is the point of no return.  You have selected to install MINIX 3
into region 0 of disk /dev/c0d0.  Please confirm that you want
to use this selection to install MINIX 3.

Are you sure you want to continue?  Please enter 'yes' or 'no': yes

--- Step 4: Reinstall choice -----

No old /home found. Doing full install.

--- Step 5: Select the size of /home -----

MINIX will take up 667 MB, without /home.
How big do you want your /home to be in MB (0-9569) ? [1913]

1913 MB Ok? [Y]

--- Step 6: Select a block size -----

The default file system block size is 4 kB.
Block size in kilobytes? [4]
```

The window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The bottom status bar shows various icons and the text "Right Ctrl".

Wait for files to be copied

Files will be automatically copied from the CD-ROM to the hard disk. Every file will be announced as it is copied.

```

minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Block size in kilobytes? [4]
You have selected to (re)install MINIX 3 in the partition /dev/c0d0p0.
The following subpartitions are now being created on /dev/c0d0p0:

  Root subpartition:  /dev/c0d0p0s0    128 MB
/home subpartition:  /dev/c0d0p0s1   1913 MB
/usr subpartition:   /dev/c0d0p0s2    rest of c0d0p0

Creating /dev/c0d0p0s0 for / ..
Creating /dev/c0d0p0s1 for /home ..
Creating /dev/c0d0p0s2 for /usr ..

--- Step 7: Wait for files to be copied -----
All files will now be copied to your hard disk. This may take a while.

Remaining: 7206 files.  [=====|-----]
/mnt/include/sys/float_ieee754.h
/mnt/include/sys/fstypes.h
/mnt/include/sys/gcqh.h
/mnt/include/sys/gmon.h
/mnt/include/sys/gmon.h
/mnt/include/sys/gpio.h
  
```

Select your Ethernet chip

You will now be asked which (if any) of the available Ethernet drivers you want installed. Network settings can be changed after installation. Please see Network Configuration (<http://wiki.minix3.org/doku.php?id=usersguide:networkconfiguration>) for details and models.


```

minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Saving random data..
1+0 records in
1+0 records out

--- Step 8: Select your Ethernet chip -----

MINIX 3 currently supports the following Ethernet cards. PCI cards detected
by MINIX are marked with *. Please choose:

0. No Ethernet card (no networking)
1. 3Com 501 or 3Com 509 based card
2. Realtek 8029 based card (also emulated by Qemu)
3. NE2000, 3com 503 or WD based card (also emulated by Bochs)
4. lan8710a (on BeagleBone, BeagleBone Black)
5. Attansic/Atheros L2 FastEthernet
6. DEC Tulip 21140A in VirtualPC
7. Intel PRO/1000 Gigabit
8. Intel PRO/100
9. * AMD LANCE (also emulated by VMWare and VirtualBox)
10. Realtek 8139 based card
11. Realtek 8169 based card
12. Virtio network device
13. Different Ethernet card (no networking)

Ethernet card? [9]

```

```

minix_R3.3.0-588a35b [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
by MINIX are marked with *. Please choose:

0. No Ethernet card (no networking)
1. 3Com 501 or 3Com 509 based card
2. Realtek 8029 based card (also emulated by Qemu)
3. NE2000, 3com 503 or WD based card (also emulated by Bochs)
4. lan8710a (on BeagleBone, BeagleBone Black)
5. Attansic/Atheros L2 FastEthernet
6. DEC Tulip 21140A in VirtualPC
7. Intel PRO/1000 Gigabit
8. Intel PRO/100
9. * AMD LANCE (also emulated by VMWare and VirtualBox)
10. Realtek 8139 based card
11. Realtek 8169 based card
12. Virtio network device
13. Different Ethernet card (no networking)

Ethernet card? [9]

Configure network using DHCP or manually?

1. Automatically using DHCP
2. Manually

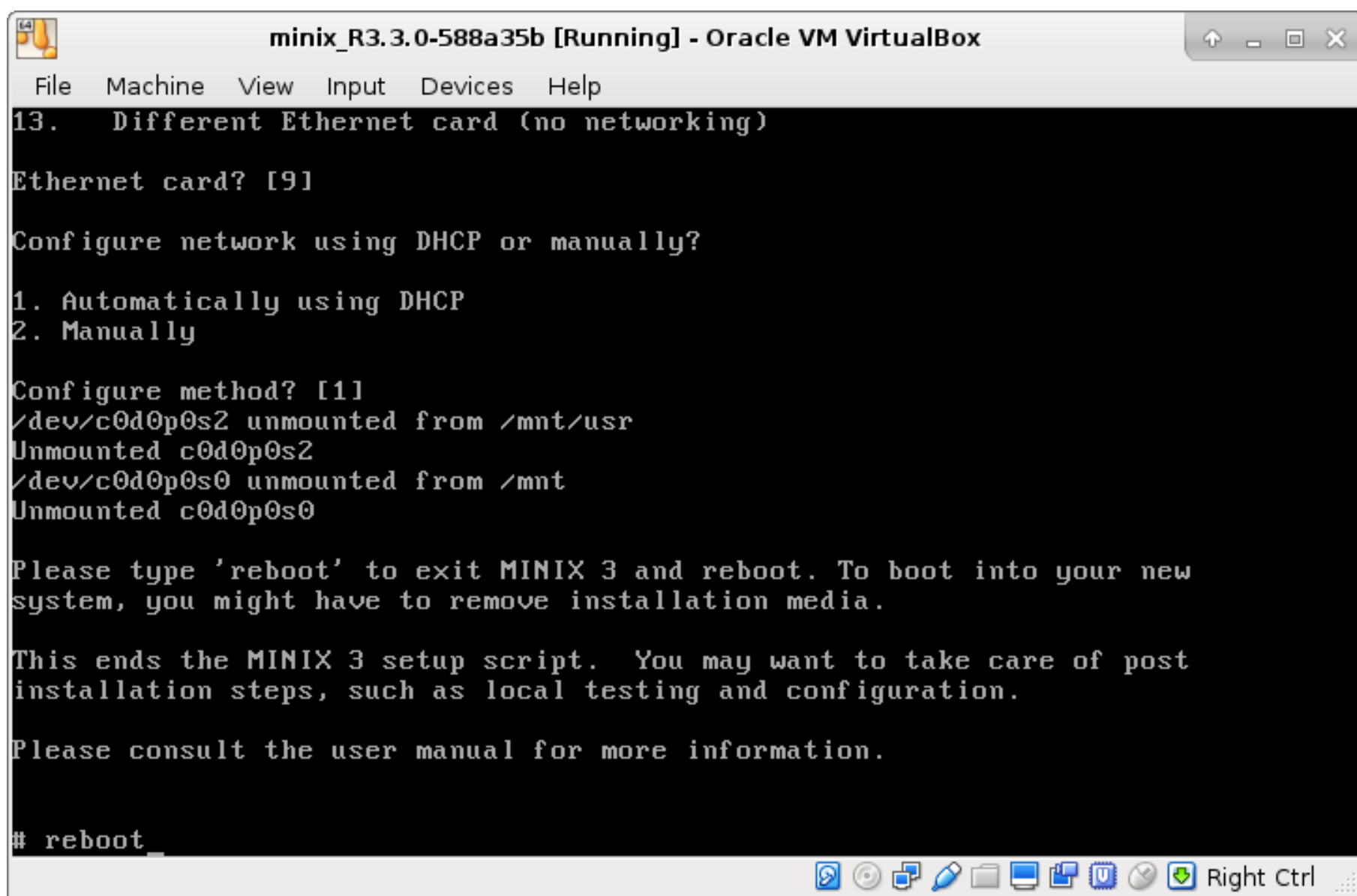
Configure method? [1]

```

Restart

When the copying is complete, MINIX is installed. Reboot the system by typing:

```
Reboot
```



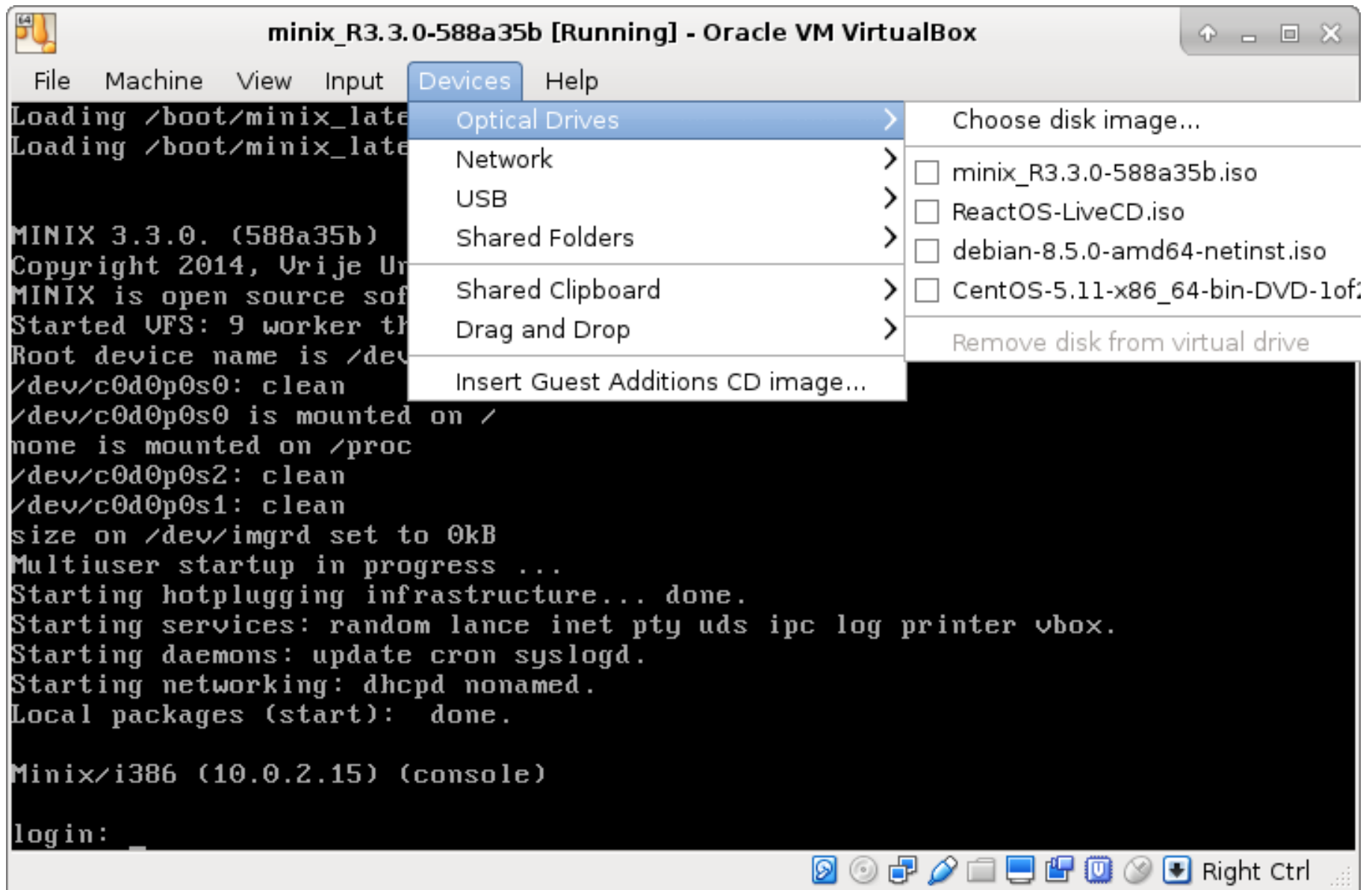
Always stop MINIX this way to avoid data loss as MINIX keeps some files on the RAM disk and only copies them back to the hard disk at shutdown time.

You can now remove any CD-ROM or floppy disk and turn off the computer or virtual drive machine. When you boot up again, you will be running MINIX.

Virtual Machines

If you are running a virtual machine, you will need to unmount the ISO image and tell the VM to boot from the hard disk. See the installation page for your VM for how to do this.

```
The virtual machine will automatically close.
```



Booting MINIX 3

Now you have installed MINIX 3 on the virtual machine. The first thing that needs to be sorted out is that next time you boot, you want to boot from the operating system, and not from the CD image.

1. Make sure your VM is selected, then click the Settings button on the main screen.
2. In the menu on the right, click on Storage.
3. In the storage tree, select the installation .iso file and click the small remove button below.
4. Great, now you can boot into the newly installed operating system.
5. Press the big Start button on the main screen.

Post-install Configuration

You should read Post Installation (<http://wiki.minix3.org/doku.php?id=usersguide:postinstallation>) for some configuration tips.

X.org

VirtualBox's guest additions are not available for MINIX3. Therefore, MINIX cannot correctly guess the screen resolution. The desired screen resolution has to be set manually in the `xorg.conf` file.

Changing screen resolution

Make sure you are not running X!

Login as root, and run the following command:

```
# Xorg -configure
```

This command should create a `xorg.conf.new` file in `/root`.

In Section “Screen” from `xorg.conf.new` file, make sure to remove all SubSection “Display”, except the one containing: Depth: 16.

Add the desired screen resolution. Possible screen resolutions can be found in

```
/var/log/Xorg.0.log.
```

Search for Modes: containing BitsPerPixel: 16, this is important!.

Example:

```
*Mode: 117 (1024x768)

[...]

XResolution: 1024

YResolution: 768
```

```
[...]
```

```
BitsPerPixel: 16
```

```
[...]
```

These resolutions can be added to the newly generated `xorg.conf.new` (in the example above: `1024x768`). I was able to use the following resolutions: `320x200`, `640x480`, `800x600`, `1024x768`, `1280x1024`, `1152x864`.

Add the desired resolution to the `Modes:` key in SubSection “`Display`”.

Example:

```
[...]
```

```
Section "Screen"
```

```
[...]
```

```
SubSection "Display"
```

```
Viewport 0 0
```

```
Depth 16
```

```
Modes "1024x768"
```

```
EndSubSection

EndSection

[...]
```

The `xorg.conf.new` file can now be moved to `/usr/pkg/X11R6/lib/X11/xorg.conf`:

```
# mv xorg.conf.new /usr/pkg/X11R6/lib/X11/xorg.conf
```

Test the new configuration file by starting X.org:

```
# startx

Sample xorg.conf

Sample xorg.conf, location: /usr/pkg/X11R6/lib/X11/xorg.conf

Section "ServerLayout"

    Identifier      "X.org Configured"

    Screen          0  "Screen0"  0  0

    InputDevice     "Mouse0"  "CorePointer"

    InputDevice     "Keyboard0" "CoreKeyboard"
```



```

EndSection

Section "Files"

    RgbPath        "/usr/pkg/X11R6/lib/X11/rgb"

    FontPath        "/usr/pkg/X11R6/lib/X11/fonts/misc/"

    FontPath        "/usr/pkg/X11R6/lib/X11/fonts/TTF/"

    FontPath        "/usr/pkg/X11R6/lib/X11/fonts/Type1/"

    FontPath        "/usr/pkg/X11R6/lib/X11/fonts/CID/"

    FontPath        "/usr/pkg/X11R6/lib/X11/fonts/75dpi/"

    FontPath        "/usr/pkg/X11R6/lib/X11/fonts/100dpi/"

EndSection

Section "Module"

EndSection

Section "InputDevice"

```

```
Identifier  "Keyboard0"

Driver      "kbd"

EndSection

Section "InputDevice"

    Identifier  "Mouse0"

    Driver      "mouse"

    Option      "Protocol" "auto"

    Option      "Device"  "/dev/mouse"

EndSection

Section "Monitor"

    Identifier  "Monitor0"
```

```
VendorName    "Monitor Vendor"

ModelName     "Monitor Model"

EndSection


Section "Device"

    Identifier  "Card0"

    Driver      "vesa"
VendorName    "Unknown Vendor"

    BoardName   "Unknown Board"

    BusID       "PCI:0:2:0"

EndSection


Section "Screen"

    Identifier  "Screen0"
```

```
Device      "Card0"

Monitor     "Monitor0"

SubSection  "Display"

            Viewport    0 0

            Depth       16

            Modes        "1152x864"

EndSubSection

EndSection
```

Port Forwarding

VirtualBox has eight networking adapters that can be separately configured to operate in one of the following six modes:

- Not attached.
- Network Address Translation (NAT).
- Bridged networking.
- Internal networking.
- Host-only networking.
- Virtual Distributed Ethernet networking.

It is possible to browse the Web, download files and view e-mail inside the guest (MINIX 3) with the Network Address Translation (<http://en.wikipedia.org/wiki/Network%20Address%20Translation>) mode. In this default mode (NAT), the guest operating system can not access the host machine or other computers on the same network and vice versa. However, like a physical router, VirtualBox can make selected services available through port forwarding (http://en.wikipedia.org/wiki/Port_forwarding). This means that VirtualBox listens to certain ports on the host and resends all packets that arrive there to the guest, on the same or a different port.

For example, to forward SSH traffic from host machine to guest machine on port 2222:

```
VBoxManage modifyvm "VM name" --natpf1 "guestssh,tcp,,2222,,22"
```

The “VM name” is the name of VM on the VirtualBox management screen, and “guestssh” is a purely descriptive name and will be auto-generated if omitted.

Connecting to guest machine with following command on host machine:

```
ssh -p 2222 localhost
```

The guest operating system is available for host machine and other machines on the network as well through the same port 2222 at the host's IP address (if host machine firewall allows it). This is useful for remote development and navigation with Eclipse Remote System Explorer (<http://wiki.minix3.org/doku.php?id=developersguide:eclipsetutorial>).

Workarounds

VirtualBox 3.1

VirtualBox 3.1 is not able to boot MINIX 3. Please use the latest version of VirtualBox.

Install issue (no hardware acceleration)

Symptom: kernel panic right after boot menu (CD loads and displays boot menu but panics right after)

Workaround:

1. If you can enable hardware acceleration:
2. Verify that your processor has the virtualization extensions (VT-x, AMD-V)
3. Enable hardware acceleration in your BIOS.

4. Go to the Settings dialog for your VM image by selecting it and clicking the Settings button on the main screen.
5. Click on System.
6. Click on the Acceleration tab.
7. Check Enable VT-x/AMD-V.
8. If you aren't able to use hardware acceleration (e.g. VirtualBox 3.1.2 + Core 2 Duo + Minix 3.2.0):
9. Follow all the installation steps as above.
10. Uncheck Enable VT-x/AMD-V.
11. Start your VM with this command: `VBoxSDL --startvm minix --norawr0 --norawr3`.
12. Replace your VM image's name for minix in the preceding command.
13. VirtualBox 4.0 has no Enable VT-x/AMD-V button, but you can issue this command to avoid kernel panics during installation: `VBoxSDL --startvm minix --norawr0 --norawr3`

DNS resolution not working

When the MINIX3 virtual machine is using (at least) NAT networking configuration, it will obtain the server address from the host system through DHCP. The VirtualBox-provided server address is the exact same address as used on the host system. On some systems, this can lead to a non-working resolution. For example, the host system uses a local resolver (on 127.0.1.1), which leads to the MINIX3 guest fruitlessly sending requests to itself rather than the host's resolver. The result is that, for example, `pkgin up` gives “Host name lookup failure” errors.

On MINIX3, the current DHCP-obtained server settings can be checked with the command `dhcpcd -q` – the server address is listed as `DNSserver`. If this address is indeed not a routable IP address, one may have to enable VirtualBox's proxy, using these instructions from the official VirtualBox website (<https://www.virtualbox.org/manual/ch09.html#nat-adv-dns>). This should resolve the issue.

Time zone issues

If you have configured a time zone in MINIX3 (for example, by putting the line “`export TZ=CET`” in `/etc/rc.timezone`), and you find that your clock (printed by, for example, the “`date`” command) ends up being ahead of real time by one or more hours, then take the following steps (tested on VirtualBox 4.1.6):

1. Shut down and power off the virtual machine (at the moment this requires a hard power-off through the VirtualBox);

2. Go to the Settings of the virtual machine;
3. Go to the System tab;
4. Under Extended features, check the “Hardware clock in UTC time” option;
5. Click on OK to save the change;
6. Restart the virtual machine, and the problem should now be fixed, even though the “wrong” (GMT) date will be printed at bootup.

Note that if your clock is behind for any reason, the MINIX3 vbox VirtualBox time sync driver will automatically correct the time for you.

Shared Folders

To use the shared folders feature please do the following:

1. Ensure the virtual machine is currently off;
2. Go to the Settings of the virtual machine;
3. Go to the Shared Folders tab;
4. Click the add button and select the folder to share from the host and assign it a name;
5. Click on OK to save the change;
6. Start the VM and login;
7. To mount your shared folder do the following:

```
mount -t vbfs -o share=NAME none /mnt
```

Be sure to replace NAME here with the name you assigned the share in step 4. Please also note that this cannot be entered into fstab for automatic mounting due to the fact that mounting takes place earlier in the boot process than the loading of the appropriate virtualbox driver for shared folders.

And to conclude, a list of some universities around the world, who use and apply their courses in the use of MINIX based knowledge to courses such as Computer Science and Computer Engineering.

University Courses Using MINIX 3

- Operating Systems Practical, Vrije Universiteit, Amsterdam, The Netherlands

- ECS 150: Operating Systems, University of California, Davis, CA, USA
- CS 170: Operating Systems, University of California, Santa Barbara, CA, USA
- CMPS 111: Introduction to Operating Systems, University of California, Santa Cruz, CA, USA
- CSCI 4730/6730: Operating Systems, University of Georgia, Athens, GA, USA
- CIS 483: Introduction to Computer & Network Security, Syracuse University, Syracuse, NY, USA
- CSE 644: Internet Security, Syracuse University, Syracuse, NY, USA
- ICS 612: Operating Systems, University of Hawaii at Manoa, Manoa, Hawaii, USA
- COMP3301/7308: Operating Systems Architecture, The University of Queensland, Brisbane, Australia
- COMP301: Operating Systems, The University of Waikato, Hamilton, New Zealand
- CMPT 507: Advanced Operating Systems, Qatar University, Doha, Qatar
- 605.412: Operating Systems, Johns Hopkins University Engineering for Professionals, Baltimore, MD, USA
- A1SO1/A1S02: Operating Systems 1/2, Federal Institute of Education, Science and Technology, Sao Paulo, SP, Brazil
- SISD: Distributed Systems, Faculty of Technoly Rubens Lara, Santos, SP, Brazil

References

<http://wiki.minix3.org/doku.php?id=www:documentation:features>

<http://wiki.minix3.org/doku.php?id=courses>

<https://www.youtube.com/watch?v=bx3KuE7UjGA>

https://en.wikipedia.org/wiki/MINIX_3

<http://wiki.minix3.org/doku.php?id=developersguide:eclipse tutorial>

<http://wiki.minix3.org/en/FrontPage>

<http://wiki.minix3.org/doku.php?id=usersguide:postinstallation>

About the Author:



Mauro Risonho de Paula Assumpção a security researcher, enthusiastic and speaker at security conferences. Is Senior Information Security Analyst (R&D) by Agility Networks, focusing on reverse engineering malware, PenTest advanced and deep research on the web, through the SIS system. He was also expert SGTI and Security Researcher (VAT) by the company ICTS Protivit, focused on R&D, VM and VA. It has focused on vulnerabilities research on Google technologies, lately in the Google Chromium source code and is also interested in

SCADA vulnerabilities, ATMs, security camera and other devices. It was the founder of the website "Backtrack Brazil" (unofficial) and from 2008 to 2012 was the moderator, participant and translator Backtrack (USA). Participates with official member of the OpenVAS project and is OWASP ZAPROXY Evangelist. Lately it has focused on open source The Penetration Testers Framework project (PTF) of trustedsec company, creator of the open source Social-Engineer Toolkit project (SET).

Linkedin: <https://br.linkedin.com/in/firebitsbr>

MINIX - A Class-based Operating System

by Rafael Santiago de Souza Netto

This first article intends to introduce the MINIX Operating System, as well as talk about some basic technical and historical aspects involved with it. Also, it will include some general details about MINIX. In addition, it will tease you learn more about Operating Systems in general.

A class-based Operating System

Operating Systems (OS) is a demanding subject not only for students but also for teachers, especially teachers who worry about the quality of their classes.

It would be so easy to turn an Operating System course into a simple boring course related to only one or a few Operating Systems.

The most important thing to expose in an Operating System course needs to be concepts. It is important to show the ways how the entire OS could be built up.

Main ideas (including algorithms) about memory management, file-systems, process management, and so on, should be explored.

However, a real life OS should face a bunch of issues in order to be efficient. This claim for efficiency makes the OS code bloat up. The usage of a real-life OS for course labs becomes hard because the code reading demands a lot from the reader. The student must know several peripheral details about some specific subject to figure out the main subject there. These peripheral details usually take the concepts far away from us.

Unfortunately, some teachers prefer to abstract too much only showing the standard commands from a specific OS. Still, some of them prefer to present an OS and its idiosyncrasies as a standard pattern followed by every OS, etc. In fact, usually these preferences produce poor courses.

As an attempt to oppose it, a teacher called John Lions took UNIX's source code (V6) and studied it, adding several useful comments. As a result, he produced lecture notes to be used in his Operating System classes. It was 1976 and these lecture notes became rather famous, producing not only several photocopies but also polemics due to the UNIX copyright's changes done years after. The UNIX V6 was essentially written by Dennis Ritchie and Kenneth Thompson, by the way, two great names in the Computer Science field.

The nice fact for using AT&T's UNIX V6 is that this version can be considered quite simple but still useful. The UNIX V6 had a good code base to expose concepts about what an OS should be, but the lack of a good and simple OS code to use during classes seemed ended up. Unfortunately, for a few years, with the copyright's changes, the code usage for classes became impossible.

Meanwhile, another teacher called Andrew S. Tanenbaum took a courageous decision. He decided to create his own UNIX clone to be used in his Operating System courses.

Tanenbaum's UNIX is called MINIX, this OS teacher also produced a well-known book about Operating Systems, using his own MINIX in technical examples about some introduced concepts.

In my opinion, UNIX V6 and MINIX are good pieces of software to be studied. Not because today it could be considered toy-OSes but because they are focused on concepts taking some complications out (like computer networks, trend-fancy-devices that everyone of us are hooked on, etc.).

This valuable property makes the study of Operating System concepts somewhat easier. Due to it, today, these Operating Systems are class-based OSes, I am sorry for the lousy pun... Anyway, in order to know more about UNIX V6, you should read the book "Lions' Commentary on UNIX 6th Ed., with source code". Today it is not prohibited anymore. If you are intending to buy it, buy the version that includes the source code. Tip: look for the sentence "with source code" in title.

Now in the following sections, I will try to show you some aspects of MINIX. Even so, you still should read Tanenbaum & Woodhull's "Operating Systems: Design & Implementation" book. With these articles, I really expect to tease you to read this book. Do not worry about spoilers!

Some facts about MINIX

Internally, MINIX differs from UNIX. MINIX was written seeking to be equal to UNIX from the user's point of view and implement the most important aspects about it. Minimal but useful.\

Another thing about MINIX is the project motivation. MINIX was written thinking about students, so you will find several commentaries along with its source code. The performance is not more important than the readability there. It opposes a production OS, which sometimes, due to requirements, needs to do tricky things. MINIX tries to contradict a famous UNIX fortune that you

may have seen during your logins:

kernel, n.:

A part of an operating system that preserves the medieval traditions of sorcery and black art.

MINIX's kernel is quite small when comparing it with other commercial/real-life OSes. According to Tanenbaum & Woodhull's book, MINIX 3's kernel has less than 4000 lines of code. By the way, the kernel architecture adopted in the MINIX project is a micro-kernel, as opposed to the main real-life UNIX-like operating systems that we have around today: FreeBSD, Linux, etc. Some BSDs implement the micro-kernel message passing but I am not so sure about saying that they adopt the entire micro-kernel philosophy.

At this point, we get an important remark here: an OS is not micro-kernel just because it allows loadable modules.

Okay, you are a user-programmer and do not know anything about OS kernel architecture. In general, try thinking about a micro-kernel architecture as a

Software Project that strongly uses dynamic linkage, and a monolithic architecture as a Software Project which links every resource statically into only one binary.

The first produces a small and spread (several binaries artifacts) code, the second one produces a huge and concentrated (only one binary) code.

Until today we have discussions about what is the best architecture between the two. Monolithic vs. Micro-kernel, the endless polemic. A classic discussion about this theme is the discussion between Tanenbaum and Linus Torvalds. You can find it easy using your web-search engine. I think that more details are unnecessary here.

The main goal of MINIX is to be a UNIX clone from the user's point of view. So the UNIX internals do not matter so much because it must be easy for the students.

Even differing from the original UNIX, with a micro-kernel, MINIX is POSIX compliant. Originally, it was written to be compatible with UNIX V7.

Did you say POSIX?

Yes, POSIX is not about another UNIX-like OS or UNIX-like OS distribution. POSIX is a standard created by IEEE to make possible the interchange of programs among the several UNIX implementations that we have. Something like:

Write once, compile and run in every "POSIXware"...

The Single Unix Specification is composed by three documents: ANSI C (Standards about our beloved C Language), XPG4 (Standards about the X Server) and the POSIX.

The POSIX is maintained by the IEEE and the Open Group in the U.S. and by the ISO/IEC in Europe.

Basically, the POSIX Standard defines the system calls that any UNIX compliant operating system should implement. It includes messages and signals shared by the processes. Table 1 gathers these signals and summarizes some points about them.

Portable Number	Name	Description
1	<u>SIGHUP</u>	Hangup
2	SIGINT	Terminal interrupt signal
3	SIGQUIT	Terminal quit signal
-	SIGILL	Illegal instruction
-	SIGTRAP	Trace/breakpoint trap
6	SIGABRT	Process abort signal
-	SIGIOT	Process abort signal. (PDP-11)
-	SIGEMT	Obsolete
-	SIGUNUSED	Unused
9	SIGKILL	Kill (can not be ignored)
-	SIGFPE	Erroneous arithmetic operation
-	SIGUSR1	User-defined signal 1
-	SIGBUS	Access to an undefined portion of a memory object
-	SIGSEGV	Invalid memory reference
-	SIGUSR2	User-defined signal 2
-	SIGPIPE	Write on a pipe with no one to read it (broken-pipe)
14	SIGALARM	Alarm clock
15	SIGTERM	Termination signal (can be ignored)
-	SIGCHLD	Child process terminated, stopped, or continued
-	SIGCONT	Continue executing, if stopped

-	SIGSTOP	Stop executing (cannot be caught or ignored)
-	SIGTSTP	Terminal stop signal
-	SIGTTIN	Background process attempting read
-	SIGTTOU	Background process attempting write

Table 1: The POSIX signals which your "YOUNix" must implement to be compliant with the POSIX Standard. These were extracted from `signal.h` from MINIX's source code.

Then, if you use the command "kill" passing the numbers listed in Table 1, you will send the related signal to a process.

For example, when we want to terminate a process:

```
_ kill -9 <pid>

or:

_ kill -SIGKILL <pid>
```

I think that C programmers tend to like the second way due to the #define-like mnemonic usage.

In fact, inconsistently we use "-9" to kill a process because we "know" that the POSIX Standard states this value for it. See? Maybe you know more about POSIX that you can realize.

The majority of the signals should be their values defined by the developer, however, some signals for portability issues must follow the value previous defined. Again, look at the SIGKILL case.

Still in Table 1, you can see something like: "can be ignored" and "can not be ignored"... Some signals, even if you are trying to explicitly ignore them, will not be ignored. Then, if you wrote this following idea into a C program:

```
#include <signal.h>

(...)
```

```
int main(int argc, char **argv) {  
  
    (...)  
  
    signal(SIGINT, SIGIGN);  
  
    return 0;  
  
}
```

Even with the explicit ignoring used above, your program still would be interrupted by a SIGINT signal. This is not a malfunction from your system or libc. It is just because your system is being POSIX compliant.

To start studying these signals can be a good way to dive into UNIX concepts, using the C Language to explore them and so on. I think that the usage of the C Language is important because maybe it can be the last frontier between the kernel-space and the user-space. So, using a C program, you will be directly using the original user's interface for these system calls.

Back to MINIX

MINIX, even being minimal when compared with other OSs, is huge to have its details treated in only one article. For this reason, in the following articles, I will seek to talk more about some parts of this system, specific features and installation issues. Nevertheless, I still want to give you some tips, in case you are intending to read the MINIX book.

Tanenbaum & Woodhull's book brings some important parts of the code in code listing form as a big appendix. The best way to read this book is reading about the theory and “see” this theory C-expressed in this appendix.

I find that an intermediate C knowledge is desirable. If you have the code reading habit it will be awesome too. If you still do not have it, this book can teach you about it. At least you have a well-organized code base for debuting yourself in the code reading “blues”. ;)

Even minimal, MINIX has device drivers, networking code stuff, and the need to follow other standards not so well organized as POSIX, but for simplicity issues, these things are not treated in its text book.

<http://minix1.woodhull.com/index1.html>

<http://minix1.woodhull.com/index.html>

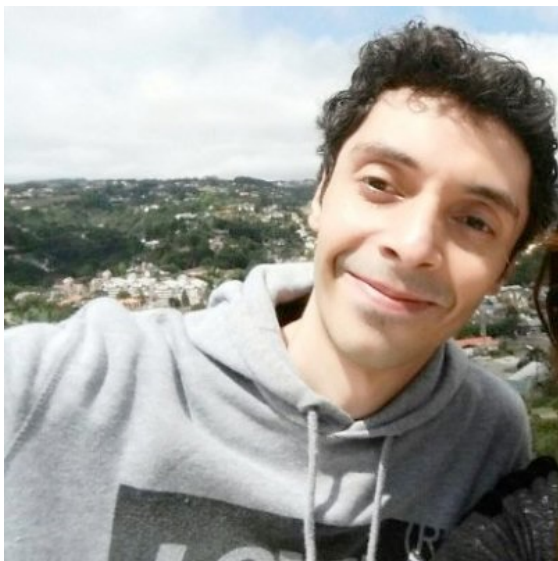
<http://www.minix3.org/>

How valuable is a good idea?

Well, after all, have spoken not only about MINIX but also about UNIX, maybe the adoption of UNIX-like systems in the early Computer Age to help train so many students along these years, could explain this idea's success. Yes, I like to think that UNIX today is more than an OS. It is a great idea about what an OS and its tools should be. Good pieces of software are not a bunch of code but a bunch of good concepts, insights and ideas. In addition, sharing could be an important way to make your ideas live forever even after you are gone. Try to ask a philosopher to summarize Plato's Theory of Forms.

In addition, you will find UNIX system calls implemented into non UNIX-like systems. Yes, sometimes it is poorly implemented but is there. Now, it is up to you to realize why.

Maybe the Time could answer the current section's title.



About the Author:

Rafael Santiago de Souza Netto is a Computer Scientist from Brazil. His main areas of interest are Programming, Computer Networks, Operating Systems, UNIX culture, Compilers, Cryptography, Information Security, Social Coding, among others. He has been working as Software Developer since 2000. You can find him at GitHub too (as rafael-santiago) where he usually 'pushes' some of his weekend projects

Optimizing In-Memory Cache of the BeaST Architecture

by Mikhail E. Zakharov

The BeaST is the new FreeBSD based dual-headed reliable storage system concept. Recently, we implemented both ZFS and in-memory cache in our architecture. After this last improvement, the BeaST system has become quite complex compared to its predecessors.*

The current BeaST version uses full-mirrored in-memory cache. In other words, all read- and write- cache partitions are mirrored between controllers. This architecture was chosen with the only aim to simplify ZFS and in-memory cache neighboring tricks.

But cache is one of the most important yet quite expensive, from multiple points of view, storage system components. Therefore, it may be a good idea to reorganize the cache architecture in order to save more resources. And the main target in the BeaST concept is to avoid unnecessary read-cache mirroring as this type of cache consumes resources but contains only not-unique data, which can be read again anytime from the drives.

Another interesting thing regarding the BeaST read-cache is related to ZFS algorithms. It appears that L2ARC is completely unused if main ARC is disabled. This case was investigated by Adam Stylinski <stylinae@mail.uc.edu>:

Just tested this and verified it for myself -- it is still the case that if no primarycache is enabled, secondary caching will not take effect.

I tested this by constructing a quick zpool in a 10-STABLE VM by using a file for a vdev, and carving out from the existing zpool a small zvol for the cache (since cache vdevs have to consist of drives or partitions).

* <https://mezzantrop.wordpress.com/portfolio/the-beast/>

I made it tiny but disabled primary cache, ran dd from /dev/urandom into files on that pool to the point where it was about 50% full, then I did sequential and random reads with grep and dd to those files, watch both the output of iostat and the output of the following systats:

```
kstat.zfs.misc.arcstats.l2_misses & kstat.zfs.misc.arcstats.l2_hits.  
  
...  
  
[adam@fbsd-stable-vm:/home/adam]%sysctl -a | grep -i arc | egrep  
"(l2_hits|l2_misses)"  
  
kstat.zfs.misc.arcstats.l2_misses: 57049  
  
kstat.zfs.misc.arcstats.l2_hits: 0  
  
[adam@fbsd-stable-vm:/home/adam]%sysctl -a | grep -i arc | egrep  
"(l2_hits|l2_misses)"  
  
kstat.zfs.misc.arcstats.l2_misses: 58751  
  
kstat.zfs.misc.arcstats.l2_hits: 0  
  
[adam@fbsd-stable-vm:/home/adam]%sysctl -a | grep -i arc | egrep  
"(l2_hits|l2_misses)"  
  
kstat.zfs.misc.arcstats.l2_misses: 61065  
  
kstat.zfs.misc.arcstats.l2_hits: 0
```

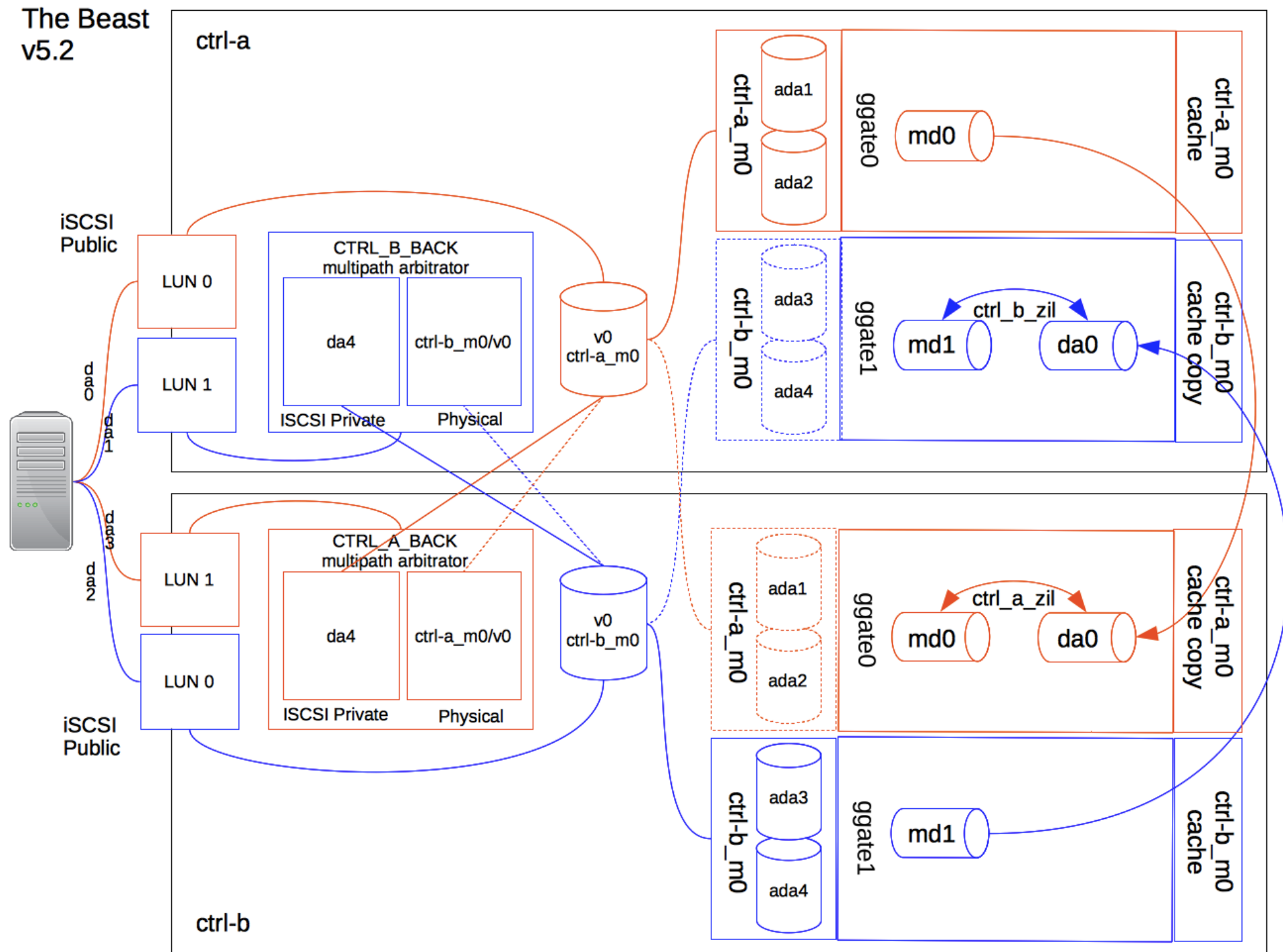
Many thanks to Adam Styliniski for his great work!

Cache architectural improvements

As for our system, it means that these memory partitions are reserved but are not used by read-cache. Therefore, we must review the architecture in order to remove read-cache mirroring between the controllers.

Being reorganized, the BeaST architecture looks much simpler and, what is more important, it allows ARC to cache data:

The Beast
v5.2



Basic preparations

To run the new version we will use exactly the same environment, which is left from the tests described in the Implementing in-memory cache in the BeaST architecture-1.1 paper. The only change is that we finally decided to drop out that slow, and thus annoying, USB-memory stick.

The configuration summary to reproduce the environment is shown below:

Description	ctrl-a	ctrl-b	clnt-1
Inter-controller (private) network. Host-only adapter (vboxnet0)	IP: 192.168.56.10 Mask: 255.255.255.0	IP: 192.168.56.11 Mask: 255.255.255.0	-
Public network. Host-only adapter (vboxnet1)	IP: 192.168.55.10 Mask: 255.255.255.0	IP: 192.168.55.11 Mask: 255.255.255.0	IP: 192.168.55.20 Mask: 255.255.255.0
Base memory	2048 MB or more	2048 MB or more	Any appropriate value starting with 512 MB will do
Shareable, fixed-sized virtual drives for ZFS data volumes on the SATA controller.	d00, d01, d10, d11 – each drive is 100 MB size or more	d00, d01, d10, d11 – each drive is 100 MB or more	-
System virtual drives (Dynamic-sized) on the IDE controller	At least 5 GB to store FreeBSD 10.3-Release default installation	At least 5 GB to store FreeBSD 10.3-Release default installation	At least 5 GB to store FreeBSD 10.3-Release default installation

Install FreeBSD 10.3 Release on the non-shareable drives (ada0 in our case) of the virtual storage machines with the typical for our project configuration changes in `/etc/rc.conf`:

ctrl-a	ctrl-b
<code>hostname="ctrl-a"</code> <code>ifconfig_em0="inet 192.168.56.10 netmask</code>	<code>hostname="ctrl-b"</code> <code>ifconfig_em0="inet 192.168.56.11 netmask</code>

ctrl-a	ctrl-b
<pre>255.255.255.0" # Inter-controller LAN ifconfig_em1="inet 192.168.55.10 netmask 255.255.255.0" # Public network sshd_enable="YES" # Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable dumpdev="AUTO" # VirtualBox guest additions vboxguest_enable="YES" vboxservice_enable="YES" # iSCSI ctld_enable="YES" # Targets iscsid_enable="YES" # Initiators</pre>	<pre>255.255.255.0" # Inter-controller LAN ifconfig_em1="inet 192.168.55.11 netmask 255.255.255.0" # Public network sshd_enable="YES" # Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable dumpdev="AUTO" # VirtualBox guest additions vboxguest_enable="YES" vboxservice_enable="YES" # iSCSI ctld_enable="YES" # target iscsid_enable="YES" # initiator</pre>

Set iSCSI “disconnection on fail” kernel variable in `/etc/sysctl.conf` on both systems to enable failover to the alive controller in case of disaster:

```
kern.iscsi.fail_on_disconnection=1
```

After finishing basic FreeBSD installations and preparations, we can start our modified in-memory cache configuration.

ZFS basic configuration

It is very simple as now we create only pools on both controllers and volumes to store data:

ctrl-a	ctrl-b
<pre>zpool create -m none ctrl-a_m0 /dev/ada1 /dev/ada2</pre>	<pre>zpool create -m none ctrl-b_m0 /dev/ada3 /dev/ada4</pre>
<pre>zfs create -V 120M ctrl-a_m0/v0</pre>	<pre>zfs create -V 120M ctrl-b_m0/v0</pre>

ZIL configuration and cross-controller pools import are described in the next section.

In-memory cache

Reflecting the changes in the cache architecture design, our memory drive to GEOM-gate map has also been changed. It is simpler now as the drives will contain only write-cache:

Memory drive	ZFS inter-layer	Controller	Description
md0	ggate0	ctrl-a	ctrl-a write-cache (ZFS ZIL) primary copy
md1	ggate1	ctrl-a	ctrl-a write-cache (ZFS ZIL) secondary copy
md0	ggate0		ctrl-a write-cache (ZFS ZIL) secondary copy
md1	ggate1	ctrl-b	ctrl-a write-cache (ZFS ZIL) primary copy

To reproduce the renewed cache structure on the system run:

ctrl-a	ctrl-b
<pre>mdconfig -a -t swap -s 128m -u 0 mdconfig -a -t swap -s 128m -u 1 ggatel create -t 1 -u 0 /dev/md0</pre>	<pre>mdconfig -a -t swap -s 128m -u 0 mdconfig -a -t swap -s 128m -u 1 ggatel create -t 1 -u 1 /dev/md1</pre>

Don't forget to load gmirror as we will need it soon:

ctrl-a	ctrl-b
<pre>gmirror load</pre>	<pre>gmirror load</pre>

Now we can prepare the iSCSI targets part for the cache synchronization mechanism in the `/etc/ctl.conf` file:

ctrl-a	ctrl-b
<pre>portal-group pg0 { discovery-auth-group no- authentication listen 192.168.56.10 } target iqn. 2016-01.local.sss.private:target0 { auth-group no- authentication portal-group pg0</pre>	<pre>portal-group pg0 { discovery-auth-group no- authentication listen 192.168.56.11 } target iqn. 2016-01.local.sss.private:target0 { auth-group no- authentication portal-group pg0</pre>

ctrl-a	ctrl-b
<pre># ctrl-a ZIL primary copy lun 0 { path /dev/md0 } }</pre>	<pre># ctrl-b ZIL primary copy lun 1 { path /dev/md1 } }</pre>

Then establish iSCSI connections:

ctrl-a	ctrl-b
<pre>service ctld start iscsictl -A -p 192.168.56.11 -t iqn. 2016-01.local.sss.private:target0</pre>	<pre>service ctld start iscsictl -A -p 192.168.56.10 -t iqn. 2016-01.local.sss.private:target0</pre>

And start mirroring processes:

ctrl-a	ctrl-b
<pre>gmirror label ctrl_b_zil /dev/ da0 /dev/md1 ggatel create -t 1 -u 1 /dev/ mirror/ctrl_b_zil</pre>	<pre>gmirror label ctrl_a_zil /dev/ da0 /dev/md0 ggatel create -t 1 -u 0 /dev/ mirror/ctrl_a_zil</pre>

Now we can enable ZIL on the in-memory mirrored drives:

ctrl-a	ctrl-b
<pre># ZIL: zpool add -f ctrl-a_m0 log /dev/ ggate0 zfs set sync=always ctrl-a_m0</pre>	<pre># ZIL: zpool add -f ctrl-b_m0 log /dev/ ggate1 zfs set sync=always ctrl-b_m0</pre>

Finally we must import both pools on both controllers and disable ZFS stop on any failure:

ctrl-a	ctrl-b
<pre>zpool import -N ctrl-b_m0 zpool set failmode=continue ctrl- a_m0 zpool set failmode=continue ctrl- b_m0</pre>	<pre>zpool import -N ctrl-a_m0 zpool set failmode=continue ctrl- a_m0 zpool set failmode=continue ctrl- b_m0</pre>

The failover arbitrator

Failover mechanism is not changed from the previous version. Let's add appropriate iSCSI target definitions to the `/etc/ctl.conf` file so it will look like:

ctrl-a	ctrl-b
<pre>portal-group pg0 { discovery-auth-group no- authentication listen 192.168.56.10 }</pre>	<pre>portal-group pg0 { discovery-auth-group no- authentication listen 192.168.56.11 }</pre>

ctrl-a	ctrl-b
<pre>target iqn. 2016-01.local.sss.private:target0 { auth-group no-authentication portal-group pg0 # ctrl-a ZIL primary copy lun 0 { path /dev/md0 } # data volumes lun 10 { path /dev/zvol/ ctrl-a_m0/v0 } } }</pre>	<pre>target iqn. 2016-01.local.sss.private:target0 { auth-group no-authentication portal-group pg0 # ctrl-b ZIL primary copy lun 1 { path /dev/md1 } # data volumes lun 10 { path /dev/zvol/ ctrl-b_m0/v0 } } }</pre>

And finally assemble the complete arbitration construction:

ctrl-a	ctrl-b
<pre>killall -HUP ctld iscsictl -M -i 1 -p 192.168.56.11 -t iqn.</pre>	<pre>killall -HUP ctld iscsictl -M -i 1 -p 192.168.56.10 -t iqn.</pre>

ctrl-a	ctrl-b
<pre>2016-01.local.sss.private:target0 gmultipath create CTRL_B_BACK / dev/dal /dev/zvol/ctrl-b_m0/v0</pre>	<pre>2016-01.local.sss.private:target0 gmultipath create CTRL_A_BACK / dev/dal /dev/zvol/ctrl-a_m0/v0</pre>

Front-end configuration

Front-end configuration is obviously simple. Change /etc/ctl.conf to add iSCSI target information for the LUNs, accessible for client-hosts. As in previous versions, we use portal-group pg1 for the public access:

ctrl-a	ctrl-b
<pre>portal-group pg0 { discovery-auth-group no- authentication listen 192.168.56.10 } portal-group pg1 { discovery-auth-group no- authentication listen 192.168.55.10 }</pre>	<pre>portal-group pg0 { discovery-auth-group no- authentication listen 192.168.56.11 } portal-group pg1 { discovery-auth-group no- authentication listen 192.168.55.11 }</pre>

ctrl-a	ctrl-b
<pre>target iqn. 2016-01.local.sss.private:target0 { auth-group no- authentication portal-group pg0 # ctrl-a ZIL primary copy lun 0 { path /dev/md0 } # data volumes lun 10 { path /dev/zvol/ ctrl-a_m0/v0 } } } } target iqn. 2016-01.local.sss.public:target0 { auth-group no- authentication</pre>	<pre>target iqn. 2016-01.local.sss.private:target0 { auth-group no- authentication portal-group pg0 # ctrl-b ZIL primary copy lun 1 { path /dev/md1 } # data volumes lun 10 { path /dev/zvol/ ctrl-b_m0/v0 } } target iqn. 2016-01.local.sss.public:target0 { auth-group no- authentication</pre>

ctrl-a	ctrl-b
<pre>portal-group pg0 portal-group pg1 lun 0 { path /dev/zvol/ ctrl-a_m0/v0 } lun 1 { path /dev/ multipath/CTRL_B_BACK } }</pre>	<pre>portal-group pg0 # ctrl-b ZIL primary copy portal-group pg1 lun 0 { path /dev/zvol/ ctrl-b_m0/v0 } lun 1 { path /dev/ multipath/CTRL_A_BACK } }</pre>

The last step is to tell ctld daemon to renew its configuration. Therefore:

ctrl-a	ctrl-b
<pre>killall -HUP ctld</pre>	<pre>killall -HUP ctld</pre>

Now we have a fully functioning dual-controlled storage system with ZFS and working in-memory cache.

You can test it with our clnt-1 client-host. The testing procedure was completely described in all past papers, therefore we will not repeat it here word for word once again.

Instead, we will think of implementing level 2 cache into the BeaST architecture, but that is a story for a future article.

Finally, our traditional warning: the BeaST is in the early development stage! It is for testing only! Do not use it in production or for storing essential data, as you can easily lose your data!



About the Author:

My name is Mikhail E. Zakharov and I am a proud SAN/storage IBMer. 10 years of experience in large SAN and storage environments: mainly Hitachi, HP and Brocade. Empty – expect-like tool author. FreeBSD enthusiast.

Fixing Failing Ports for Hardened/LibreBSD

by Bernard Spil

HardenedBSD ran an exp-run with LibreSSL in base. This was expected to uncover a lot of issues where ports check the `OPENSSL_VERSION_NUMBER` to determine if a feature is available. To my surprise, it only uncovered 12 ports that failed due to these version checks.

Prelude

The LibreSSL ports (up to 2.4) on FreeBSD include a patch that modifies the OpenSSL version in the header files:*

```
--- include/openssl/opensslv.h.orig      2015-09-11 22:35:14 UTC
+++ include/openssl/opensslv.h

@@ -7,7 +7,7 @@

#define LIBRESSL_VERSION_TEXT  "LibreSSL 2.3.0"

/* These will never change */

-#define OPENSSL_VERSION_NUMBER 0x20000000L
+#define OPENSSL_VERSION_NUMBER 0x1000107fL

#define OPENSSL_VERSION_TEXT  LIBRESSL_VERSION_TEXT

#define OPENSSL_VERSION_PTEXT " part of " OPENSSL_VERSION_TEXT
```

**FreeBSD ports no longer contain the `OPENSSL_VERSION_NUMBER` patch since last week*

This patch locks the OpenSSL version that is exposed to software to 1.0.1g in line with the forking of LibreSSL from OpenSSL.

This version modification was added to LibreSSL by the original maintainer to circumvent the issues that would arise if ports check OPENSSL_VERSION_NUMBER as a surrogate to detect features. This is a problematic way of checking for features, how will we ever be able to remove features this way!

Result of exp-run

When replacing OpenSSL with LibreSSL for HardenedBSD, I decided to do away with this change and see where I'd end up. Interestingly, only 12 ports were failing due to these checks. As more and more software starts using features from 1.0.2 and 1.1.0, this may increase, but at least the rate of these issues arising will be lower.

Port:

```
dns/bind910
ftp/curl
mail/postfix
mail/postfix-current
net/haproxy-devel
net-mgmt/send
security/openvpn
security/stunnel
security/wpa_supplicant
security/xca
```

A side-effect of this exp-run is that we are detecting ports that do not set or honor `USE_OPENSSL= yes` in the port's Makefile. This means that they weren't failing when `WITH_OPENSSL_PORT= yes` and `OPENSSL_PORT= security/libressl-devel` is set during build of ports but they are failing now because there's no OpenSSL `libcrypto/libssl` available on the system.*

**WITH_OPENSSL_PORT= yes and OPENSSL_PORT= security/libressl-devel have been replaced by DEFAULT_VERSIONS+= ssl=libressl-devel*

Port	Problem
benchmarks/postal	SSLv3
databases/mongodb32-tools	SSLv3
databases/mongodb32	SSLv3
devel/tcl-trf	SHA-0
finance/openhbc	DES_
mail/emailrelay	SSLv3
mail/mixmaster	EGD
mail/libesmtp	DES
mail/prayer	SSLv3 EGD
misc/smssend	SSLv3
multimedia/oscam	SSLv3
net/Socket	SSLv3
net/l4ip	EGD
net/netatalk	DES_
net/netatalk3	DES_
net/ssl-tunnel-client	DES_
net-mgmt/snmp++	DES_
net-p2p/shx	EGD
security/certificate-transparency	CMS

Port	Problem
<code>security/distcache</code>	SSLv3
<code>security/dsniff</code>	DES_
<code>security/rcracki_mt</code>	DES_
<code>www/tomcat-native</code>	SSLv3

All in all, I created patches for all of these issues. You can find them in LibreSSL Ports and No-SSLv3.

Statistics

All in all, there are 204 ports with issues, most have patches as well. Not sure if I'll ever get around to updating the number of fixes and the number of ports fixed as well, this is becoming increasingly complex to track using a wiki page!

Problem	Description	Number of ports
SSLv3	SSLv3 methods removed from LibreSSL 2.3	85
EGD	RAND_egd methods removed from LibreSSL	38
DES	deprecated des_ methods (replaced by DES_ methods)	29
COMP	SSL compression removed from LibreSSL	10

Problem	Description	Number of ports
SHA-0	SHA-0 methods removed from LibreSSL 2.3	8
SSLv2	SSLv2 methods removed from LibreSSL	7
arc4rand	Conflict with FreeBSD/LibreSSL libs	4
PSK	Pre-Shared Key removed from LibreSSL	4
CMS	Deprecated S/MIME methods	3
GOST	GOST methods removed	2
Other	Non categorized	25

The majority of issues is with the removal of SSLv3. This should improve quickly over the coming months as OpenSSL 1.1 gets released, which removes SSLv3 in the default build configuration as well.

Find more info here:

- <https://github.com/hardenedsd/hardenedsd> has 3 branches that default to LibreSSL in base hardened/current/master-libressl, hardened/11-stable/master-libressl and hardened/10-stable/master-libressl
- PC-BSD (TrueOS desktop) has branches drm-next-4.6 and drm-next-4.7 that use LibreSSL in base
- <https://github.com/Sp1/LibreBSD> has patch-sets for 10-stable and 11.0-RC1

You may want to cherry-pick some stuff from

<https://brnrd.eu/libressl/2016-03-05/libressl-in-hardenedsd-base-part-i.html> and <https://brnrd.eu/libressl/2016-03-06/libressl-in-hardenedsd-base-part-ii.html> as well.



About the Author:

I've used FreeBSD since version 5.x and have been active on irc for a long time and never thought that I could actually contribute much to the project. Initially, I submitted PRs for things that were broken for me (one maybe two every year). Later on, I started submitting PRs including patches to fix the problem (as a non-committer, that's very much appreciated) but again very few per year.

As I got more sophisticated in fixing things in ports, the number of PRs and patches increased.

At some point, I decided that the MariaDB 10.0 port was due. So I started copying the 5.5 port, and failing time and again to get it to work, but ultimately hacked it to build with 10.0! Along the way, I interacted with the MariaDB community to solve some of the issues and after a while it was added to Ports. Suddenly, I was a port maintainer (scary!).

In this time-frame Kubilay (koobs) Kocak enlisted me in his wiki-army.

The LibreSSL thing

I was using LibreSSL about as soon as the Portable version was released and added to ports. This required me to patch some ports (Apache, Python, ...) so they would build and run with LibreSSL. That got noted and some guys on IRC were nagging and motivating (and helping!) me to do more patches. At some point, Ken Moore (PC-BSD, BSDNow.tv) reached out to me because he wanted to do an EDGE (cutting edge PC-BSD) build with LibreSSL for ports. This was something I was looking for as it would surface all, well... most, I was later to find out, problems with using LibreSSL as libssl/crypto provider. That was a very intense couple of weeks where a poudriere run would uncover problems with LibreSSL, after patching these problems more problems would surface, etc. Most issues could be binned into categories (EGD removal, deprecated des_ methods, SSLv2 removal). All this resulted in https://wiki.freebsd.org/LibreSSL/Ports#PC-BSD_10.1.2_ports_build and a large load of patches for ports as PRs in BugZilla.

When the initial fixing and patch creation was done, Kubilay spurred me on to upstream the patches, which resulted in quite some changes, usually small, to all kinds of Open Source projects. And boy am I proud of the trivial changes that made it into these upstream projects! I can now truthfully say that changes I supplied to Python are part of software running on many millions of systems. It's only me that knows that that's factually untrue, right?

After summer, I was contacted by the LibreSSL devs from OpenBSD and asked if I'd be willing to come to their LibreSSL Hackathon in Croatia 3 weeks later. That was an intense and fruitful week out there with some great guys! Exchanging information on how LibreSSL is used "in the wild", what challenges that poses and learning on the development of LibreSSL.

Know what's awesome about spending all the effort? It's so immensely appreciated even though you don't often hear that directly. Sitting chatting in the hotel lounge during EuroBSDcon you suddenly hear the guy behind you shout out "What?!? YOU are that LibreSSL guy!?!".

Source of the article:

<https://brnrd.eu/libressl/2016-04-17/fixing-failing-ports-for-hardenedlibrebsd.html>

Deploy Docker Swarm Cluster on One Host

by Nan Xiao

Sometimes, you just want to learn the internal mechanics of Docker Swarm, but, unfortunately, there is only one Linux box at hand, and you don't want to bother to install Virtual Machines on it. In this scenario, you certainly can build a Docker Swarm cluster on one host, and this tutorial will provide a detailed guide.

1. Make sure the Go environment has been ready on your system

If not, please follow this document to setup it. Also remember add `$GOPATH/bin` into `$PATH` environment variable.

2. Install Docker Swarm:

```
# go get -u github.com/docker/swarm
```

Execute swarm command to check whether Docker Swarm is well equipped:

```
# swarm

Usage: swarm [OPTIONS] COMMAND [arg...]

A Docker-native clustering system

Version: 1.2.3 (HEAD)
```

Options:

```
--debug                debug mode [$DEBUG]

--log-level, -l "info"  Log level (options: debug, info,
warn, error, fatal, panic)

--experimental          enable experimental features

--help, -h              show help

--version, -v           print the version

.....
```

3. Modify the Docker configuration file

E.g., on my RHEL 7, the file is

```
/etc/sysconfig/docker:

# systemctl show docker

.....

EnvironmentFile=/etc/sysconfig/docker (ignore_errors=yes)

.....
```

Add “-H tcp://127.0.0.1:2375” in OPTIONS field:

```
# cat /etc/sysconfig/docker

# /etc/sysconfig/docker

# Modify these options if you want to change the way the docker dae-
mon runs

OPTIONS='--selinux-enabled -H tcp://127.0.0.1:2375 -H
unix:///var/run/docker.sock'
```

Restart Docker, and check whether the new OPTIONS takes effect:

```
# systemctl restart docker

# systemctl status docker

• docker.service - Docker Application Container Engine

   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled;
   vendor preset: disabled)

   Active: active (running) since Wed 2016-06-08 12:32:19 CST; 10s
   ago

     Docs: http://docs.docker.com

  Main PID: 14429 (sh)

    CGroup: /system.slice/docker.service

            └─14429 /bin/sh -c /usr/bin/docker-current daemon $OPTIONS
              $DOCKER_STORAGE_OPTIONS                $DOCKER_NETWORK_OPTI...

            └─14430 /usr/bin/docker-current daemon --selinux-enabled
-H tcp://127.0.0.1:2375 -H unix:///var/run/docker.sock --add-
registr...

            └─14431 /usr/bin/forward-journald -tag docker

.....
```

4. Run “swarm create” command to create token for the cluster:

```
# swarm create

d10eacbda9763b0740548a2a4c2f1a59
```

5. Execute swarm join to create a Docker Swarm node:

```
# swarm join --addr 127.0.0.1:2375
token://d10eacbda9763b0740548a2a4c2f1a59

INFO[0000] Registering on the discovery service every 1m0s...
```

```
addr=127.0.0.1:2375
discovery=token://d10eacbda9763b0740548a2a4c2f1a59

.....
```

You should notice that the argument of `--addr` option is the IP and port of the Dockerengine on this host. Since we have set the `OPTIONS` in Docker configuration file in step 3, the IP should be 127.0.0.1 whilst port is 2375.

6. Open a new terminal, and create the manager of the cluster

```
# swarm manage -H 127.0.0.1:3375
token://d10eacbda9763b0740548a2a4c2f1a59

INFO[0000] Listening for HTTP
                        addr=127.0.0.1:3375 proto=tcp

INFO[0001] Registered Engine localhost.localdomain at 127.0.0.1:2375
```

Because port 2375 is occupied by Docker engine, we use another available port:

Through the log, you can see the node and manager have communicated successfully.

Now, you can think a Docker engine is listening on `tcp://127.0.0.1:3375`, but actually, there is one Docker cluster behind `tcp://127.0.0.1:3375`, even though the cluster has only one

```
# docker -H tcp://127.0.0.1:3375 info

Containers: 0

Images: 5

Server Version: swarm/1.2.3

Role: primary

Strategy: spread

Filters: health, port, containerslots, dependency, affinity, constraint
```

```
Nodes: 1

localhost.localdomain: 127.0.0.1:2375

  L ID: ZUIV:BMPV:3B5R:2WBC:JXEI:2S6H:XM3H:66W5:UZQI:NJON:JY4T:HIFB

  L Status: Healthy

  L Containers: 0 (0 Running, 0 Paused, 0 Stopped)

  L Reserved CPUs: 0 / 8

  L Reserved Memory: 0 B / 12.1 GiB

  L Labels: executiondriver=native-0.2,
kernelversion=3.10.0-327.el7.x86_64, operatingsystem=Red Hat Network,
storage driver=devicemapper

  L UpdatedAt: 2016-06-08T04:58:05Z

  L ServerVersion: 1.9.1

Kernel Version: 3.10.0-327.el7.x86_64

.....
```

Or run a container:

```
# docker -H tcp://127.0.0.1:3375 run hello-world

Hello from Docker.

This message shows that your installation appears to be working cor-
rectly.

Nodes: 1
```

To generate this message, Docker took the following steps:

```
1. The Docker client contacted the Docker daemon.
```

```
.....
```



About the Author:

My name is Nan Xiao, a system software engineer from China. I like researching and hacking the

infrastructure related technology of computer science, such as Operating System, debugging, tracing,

C/Go programming languages, etc. In my spare time, I also write some posts and thoughts about technology, and hope these articles can help others!

<http://nanxiao.me/en/deploy-docker-swarm-cluster-on-one-host/>

Using ZFS to Fight Data Rot

by Kevin McAleer

Previously, I wrote an article for BigAdmin about why I chose the ZFS file system to ensure my data was safe: “How I Used Solaris OS and ZFS to Solve My Mac OS X Storage Problem.”*

One of the reasons I chose the ZFS file system as opposed to Apple HFS+, Linux ext3/ext4, or Microsoft Windows NTFS is because the ZFS file system checksums all the data written to and read from it. This might seem unnecessary, a little obsessive, or even CPU-hungry, but it is essential for long-term data storage and for detecting data rot.

On Windows Server 2012, you can choose to use ReFS, which has some of the functionality of ZFS, such as checksumming and copy-on-write, however, it doesn't currently do deduplication or compression like ZFS.

So what is data rot, why should I fear it, and most importantly, what can I do about it?

Quite simply, data rot is the result of tiny changes in the magnetic particles that make up the media in hard disks, it may also be caused by faulty memory cells on SSD disks. The effect this has on your data is random but predictable: data loss. It might be the contents of a file that gets corrupted, the file header that describes the contents of the file, or, worse, the file allocation table that describes the location or links to the file. The file might be a system file or a data file; either way, it's eventually going to be bad news.

According to a recent study, Analyzing the Effects of Disk-Pointer Corruption (pdf), 0.66% of SATA disks and 0.06% of Fibre Channel disks developed corruption in 17 months of use. The same article describes how some corruption is worse than others and explains that most modern filing systems are unable to deal effectively with this (excluding the ZFS file system, of course!).*

http://web.archive.org/web/20090130012930/http://www.sun.com/bigadmin/content/submitted/zfs_mac_os.x.jsp

<http://web.archive.org/web/20140131190051/http://www.cs.wisc.edu/wind/Publications/pointer-dsn08.pdf>

So you're probably thinking "Doesn't chkdsk detect and correct this kind of problem (or the fsck utility or Disk Utility in Linux or Mac OS X, respectively)"? Well, maybe, maybe not, depending on where the corruption occurs. If the corruption occurs in the file system structure, then see the References* listed below. If it occurs in the file content, then the answer is "probably not".

We've established what data rot is and how existing tools are not suited to detecting, correcting, or preventing it. Now, on to why you should care about this...

How important is your data? I mean, really? Think about it. I personally have the following data stored on my computer: photos and videos of my daughter since birth, software downloads I've purchased (including Adobe Photoshop and Adobe Dreamweaver, which weren't cheap), my iTunes library (for which I must have spent a couple of hundred, if not into the triple 0's, of dollars), and various work projects.

I'm not prepared to let anything happen to this data. So I've taken steps to avoid obvious problems:

- The file server is a dedicated box.
- My data is separated out to avoid accidental deletion.

I back up my data regularly (on the Mac with Time Machine and on FreeBSD with the ZFS snapshots, which I send to an off-site duplicate via the ZFS send and receive commands). I've also taken steps to design my storage solution correctly: I use several disks in a RAID configuration (RAID-Z with a hot spare) to ensure a single disk failure can't cause data loss.

Finally, I choose to use the ZFS file system because I know that it checksums every read and write to the filing system, ensuring that my data is as it was when it was written to disk.

I run a "scrub" of the ZFS file system every week to ensure that no data has become corrupted by data rot, and this week, it detected over 20 instances of it. Thankfully, ZFS effortlessly replaced the corrupted data with good data held elsewhere on disk (thanks to RAID-Z) without any loss whatsoever.

Conclusion: To prevent data rot, choose the ZFS file system.

Although I didn't lose data, the experience did drive me to write this article, because I wanted to make people aware of this issue. I've been successfully using ZFS since its first release on Solaris in 2005, providing 11 years of data protection.

*http://web.archive.org/web/20090228135946/http://www.sun.com/bigadmin/content/submitted/data_rot.jsp#References

References

- ***An Analysis of Data Corruption in the Storage Stack*** (pdf); L. N. Bairavasundaram, G. R. Goodson, B. Schroeder, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. In FAST '08, 2008.*
- ***Analyzing the Effects of Disk-Pointer Corruption*** (pdf); Lakshmi N. Bairavasundaram, Meenali Rungta, Nitin Agrawal, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau, and Michael M. Swift. In FAST '08, 2008.*

*http://web.archive.org/web/20080706010925/http://www.usenix.org/events/fast08/tech/full_papers/bairavasundaram/bairavasundaram.pdf

*<http://web.archive.org/web/20140131190051/http://www.cs.wisc.edu/wind/Publications/pointer-dsn08.pdf>

About the Author:

Kevin McAleer is the director of Advice Factory, offering advice and IT consultancy services to businesses in the UK. He is an Apple Mac fan and also an evangelist for Oracle's ZFS technology.

FREENAS MINI STORAGE APPLIANCE

IT SAVES YOUR LIFE.



HOW IMPORTANT IS YOUR DATA?

Years of family photos. Your entire music and movie collection. Office documents you've put hours of work into. Backups for every computer you own. We ask again, *how important is your data?*

NOW IMAGINE LOSING IT ALL

Losing one bit - that's all it takes. One single bit, and your file is gone.

The worst part? **You won't know until you absolutely need that file again.**



Example of one-bit corruption

THE SOLUTION

The FreeNAS Mini has emerged as the clear choice to save your digital life. **No other NAS in its class offers ECC (error correcting code) memory and ZFS bitrot protection to ensure data always reaches disk without corruption and *never degrades over time.***

No other NAS combines the inherent data integrity and security of the ZFS filesystem with fast on-disk encryption. No other NAS provides comparable power and flexibility. The FreeNAS Mini is, hands-down, the best home and small office storage appliance you can buy on the market. **When it comes to saving your important data, there simply is no other solution.**

The Mini boasts these state-of-the-art features:

- 8-core 2.4GHz Intel® Atom™ processor
- Up to 16TB of storage capacity
- 16GB of ECC memory (with the option to upgrade to 32GB)
- 2 x 1 Gigabit network controllers
- Remote management port (IPMI)
- Tool-less design; hot swappable drive trays
- FreeNAS installed and configured



<http://www.iXsystems.com/mini>



FREENAS CERTIFIED STORAGE



With over six million downloads, FreeNAS is undisputedly *the* most popular storage operating system in the world.

Sure, you could build your own FreeNAS system: research every hardware option, order all the parts, wait for everything to ship and arrive, vent at customer service because it *hasn't*, and finally build it yourself while hoping everything fits - only to install the software and discover that the system you spent *days* agonizing over **isn't even compatible**. Or...

MAKE IT EASY ON YOURSELF

As the sponsors and lead developers of the FreeNAS project, iXsystems has combined over 20 years of hardware experience with our FreeNAS expertise to bring you FreeNAS Certified Storage. **We make it easy to enjoy all the benefits of FreeNAS without the headache of building, setting up, configuring, and supporting it yourself.** As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS.

Every FreeNAS server we ship is...

- » Custom built and optimized for your use case
- » Installed, configured, tested, and guaranteed to work out of the box
- » Supported by the Silicon Valley team that designed and built it
- » Backed by a 3 years parts and labor limited warranty

As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS. **Contact us today for a FREE Risk Elimination Consultation with one of our FreeNAS experts.** Remember, every purchase directly supports the FreeNAS project so we can continue adding features and improvements to the software for years to come. **And really - why would you buy a FreeNAS server from *anyone* else?**



FreeNAS 1U

- Intel® Xeon® Processor E3-1200v2 Family
- Up to 16TB of storage capacity
- 16GB ECC memory (upgradable to 32GB)
- 2 x 10/100/1000 Gigabit Ethernet controllers
- Redundant power supply

FreeNAS 2U

- 2x Intel® Xeon® Processors E5-2600v2 Family
- Up to 48TB of storage capacity
- 32GB ECC memory (upgradable to 128GB)
- 4 x 1GbE Network interface (Onboard) - (Upgradable to 2 x 10 Gigabit Interface)
- Redundant Power Supply

<http://www.iXsystems.com/storage/freenas-certified-storage/>




FreeNAS Getting Started Guide: Part 3, Manual Configuration


by Mark VonFange


This article series is intended to serve as an introductory guide to assist FreeNAS users in planning, installation, configuration and administration for their FreeNAS storage systems. This month's article will cover basic configuration and administration tasks within the FreeNAS User Interface.


Setting up users and groups


One of the first things you will want to do once your FreeNAS system is up and running is to add any users or groups you will need beyond the default options. You can do this either with the sidebar navigation menu or the top bar menu. Simply click go to the Account menu, then select either "Groups" or "Users" and click on "Add Group" or "Add User". This will bring up a pop-up menu (Fig 1) to enter information.


Account

System

Tasks

Network

Storage

Directory

Account

Groups

Users

Add Group

Group ID	Group Name
1001	work
0	wheel
1	daemon
2	kmem
3	sys
4	tty
5	operator
6	mail
7	bin
8	news
9	man
13	games
14	ftp
20	staff
22	sshd
25	smmisp
26	mailnull
31	guest

<

Members

Modify Group

Delete Group

Figure 1: Adding a Group

Add User

User ID:

1004

Username:

Create a new primary group for the user:

☒

Primary Group:

Create Home Directory In:

/nonexistent

Browse

Shell:

cah

Full Name:

E-mail:

Password:

Password confirmation:

Disable password login:

☐

Lock user:

☐

Permit Sudo:

☐

Microsoft Account:

☐

SSH Public Key:

Auxiliary groups:

Available

work
_dhcp
_pflogd
audit
authpf
avahi

<<
>>

Selected

OK

Cancel

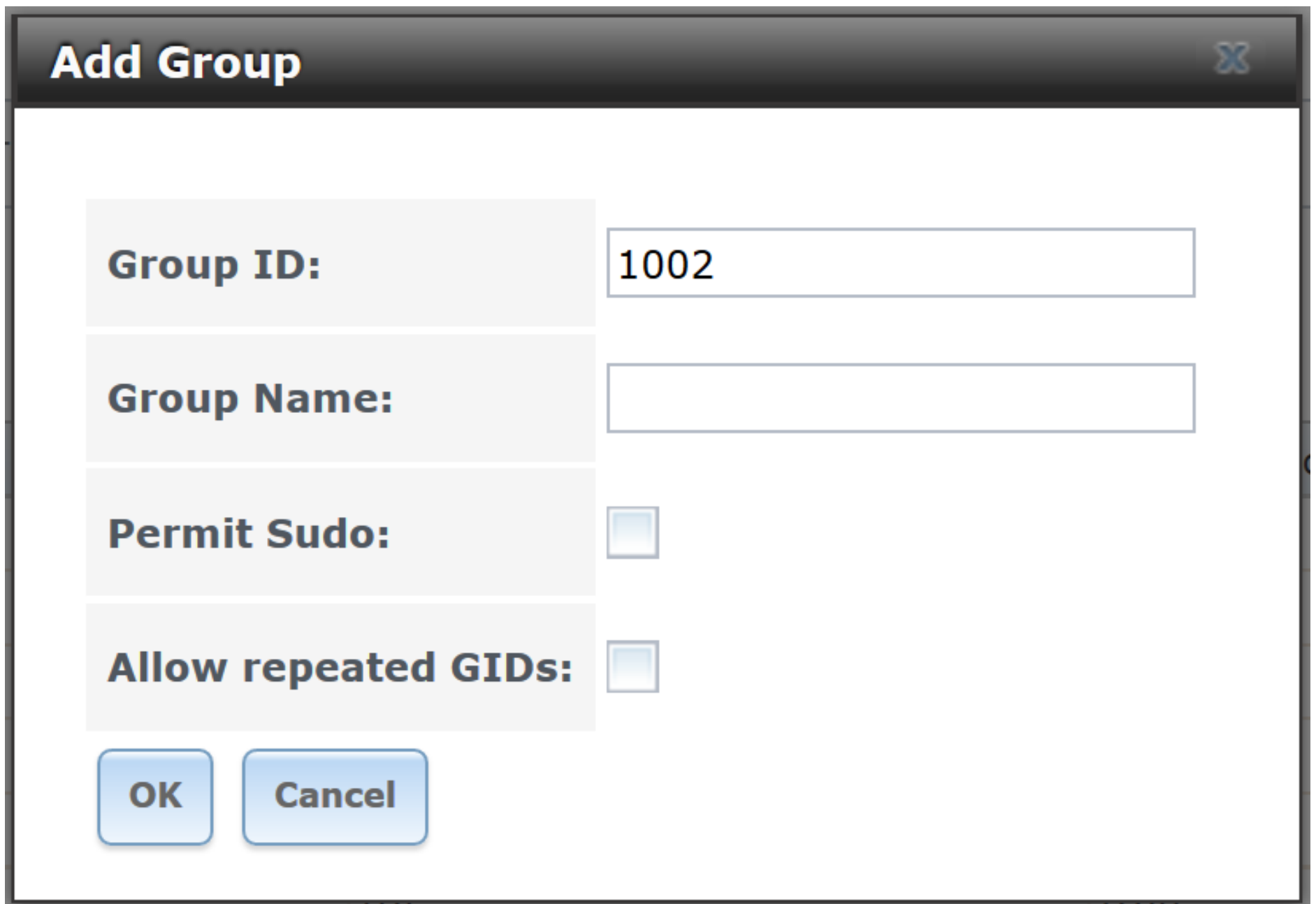
Advanced Mode

Figure 2: Add User menu

The Add User menu contains fields for your user ID, username, whether to set up a new group for this user or to add to an existing group, your preferred directory location, preferred shell for command line, your name, desired email address, and password (if desired). You can also disable passwords for this user, allow sudo access, add any additional groups you'd like the user to be a part of in the Auxiliary Groups section, and more. Once you've entered all your information, click on the OK button.

For full documentation on the Add User menu, go to https://doc.freenas.org/9.3/freenas_account.html#users.

The Add Group menu (Fig 3) will contain fields for your Group ID, Group Name, whether to allow 'sudo' access and whether to allow multiple groups to use the same Group ID (GIDs). Once you've entered all your information, just click on the "OK" button.



Add Group

Group ID: 1002

Group Name:

Permit Sudo: ☐

Allow repeated GIDs: ☐

OK **Cancel**

Figure 3 : Add Group menu

For full information on this menu, go to the documentation at https://doc.freenas.org/9.3/freenas_account.html#groups.

Once your users and groups are created, you can modify or delete them by clicking on them in the left navigation sidebar or by going to the Accounts tab, selecting the desired group or user and then clicking the “modify group” or “modify user” button at the bottom. You can also add or remove members of groups via the “Members” button in the Accounts>Groups tab.

You can find a good overview of Permissions settings on the FreeNAS Team’s YouTube channel at <https://www.youtube.com/watch?v=RBszScnsRgY>.

Volumes

While you can set up your volume in the Initial Configuration Wizard, you may wish to add it manually or extend an existing volume. To set up your volume, simply go to the “Volumes” section of the Storage menu and click on the Volume Manager button (Figure 4). First, you will need to enter a Volume Name, then you will need to choose from your available disks or select an existing Volume to extend.

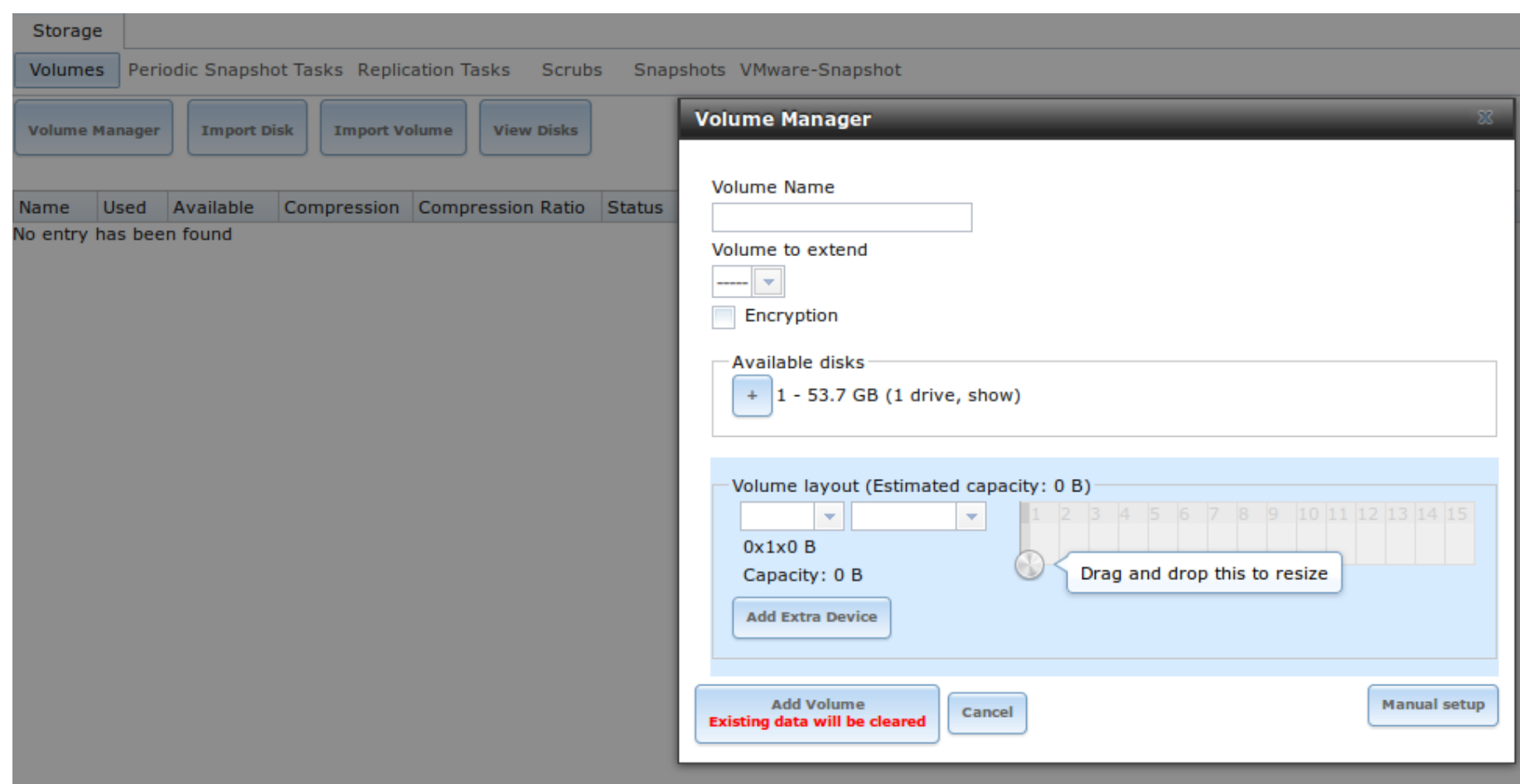
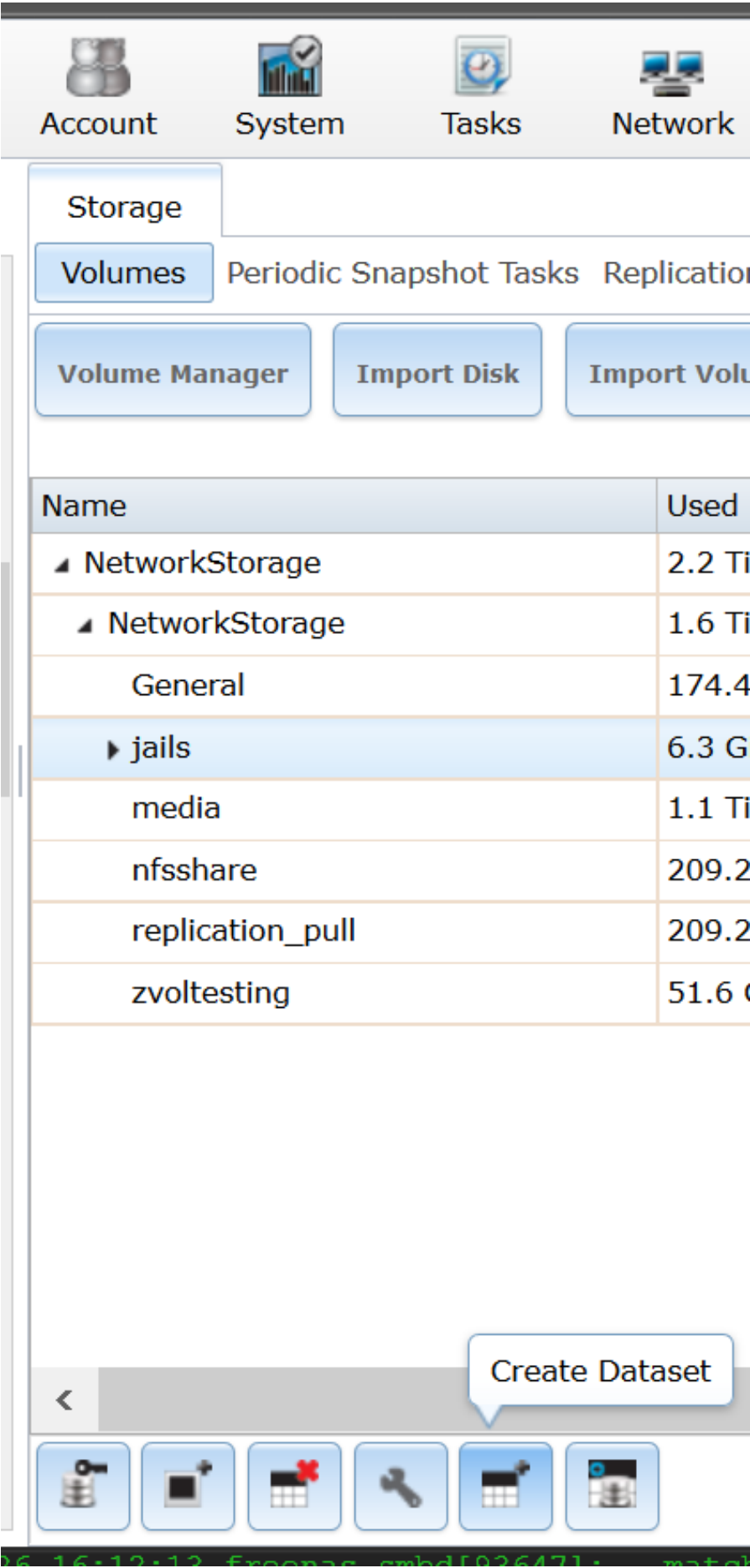


Figure 4: Volume Manager

Next, you will want to choose your volume layout via the drop down menus. ZFS has options for Stripe, Mirror and parity options with RAID-Z, RAID-Z2 and RAID-Z3.

Parity options allow for up to one, two or three drives to fail, respectively, without data loss. If you are setting up an L2ARC Read Cache or SLOG (Separate ZFS Intent Log) Write Cache devices, the volume layout drop down menu also has these options. You can utilize the drag and drop section to switch between parity and mirrored volume configurations. When finished, just click on the “Add Volume” button to set up your volume.

For further information, go to the FreeNAS documentation at https://doc.freenas.org/9.3/freenas_storage.html?highlight=volume#volume-manager. You can also check out the video on Volumes at <https://www.youtube.com/watch?v=yxnJH-8YvC8>.



Datasets and Shares

In order to enable file sharing across your network, you first need to set up Datasets. To do this, go to your desired volume from the Storage section of your sidebar navigation or the top navigation bar and select it. From the sidebar, you will need to expand the volume by pressing the ‘+’ button, then click on “Create Dataset”. From the top bar, click on the “Create Dataset” icon at the bottom of the Dataset table (Fig 5).

Figure 5: Creating a Dataset

Once you've clicked on Create Dataset, a pop-up window will open (Fig 6). The menu will have fields for your Dataset's name, desired compression level, share type and a few other options. If you click on the "Advanced Mode" button, you can set up quotas and reserved space. Quotas set the maximum amount of data capacity the dataset can use, reserved space guarantees a minimum amount of data capacity for the dataset.

Create Dataset

Create ZFS dataset in NetworkStorage

Dataset Name:

Compression level:

Share type:

Case Sensitivity:

Enable atime:

- ☒ Inherit (on)
- ☐ On
- ☐ Off

ZFS Deduplication:

Enabling dedup may have drastic performance implications, as well as impact your ability to access your data. Consider using compression instead.

Figure 6: Dataset Creation menu

Share types will correspond to the type of share you plan on using. For NFS, select UNIX, for CIFS/SMB, select Windows, and for Mac, use AFP. If you are in a mixed OS environment, you may want to use Windows (CIFS/SMB) in order to avoid share type conflicts. For full documentation of the dataset menu, go to https://doc.freenas.org/9.3/freenas_storage.html#create-dataset.

Once you've created your dataset, you can now create a share. Go to your Sharing menu and select the section for the type of share that you would like to create (this should correspond to the share type you listed for the dataset), then click the Add Share button (Fig 7). First thing you will need to do is set up the Path for the share, which will bring up a pop-up menu (Fig 8). Click the

browse button to navigate to the desired directory location. For Apple, WebDav and Windows you will need to name your share. For Apple shares, you can set your share up for TimeMachine back-ups. For Windows shares, select “Allow Guest Access” if you do not want to require a password.

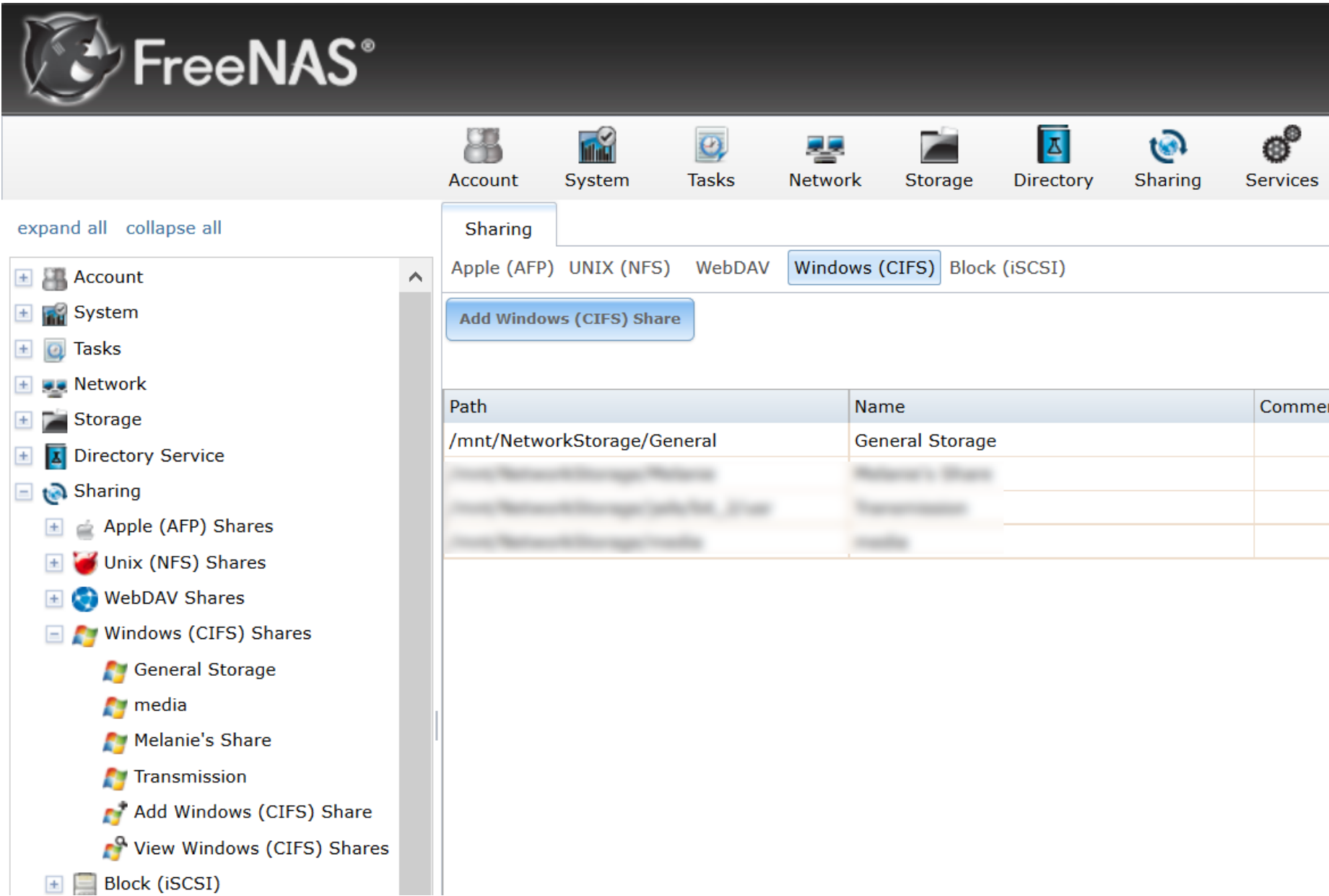
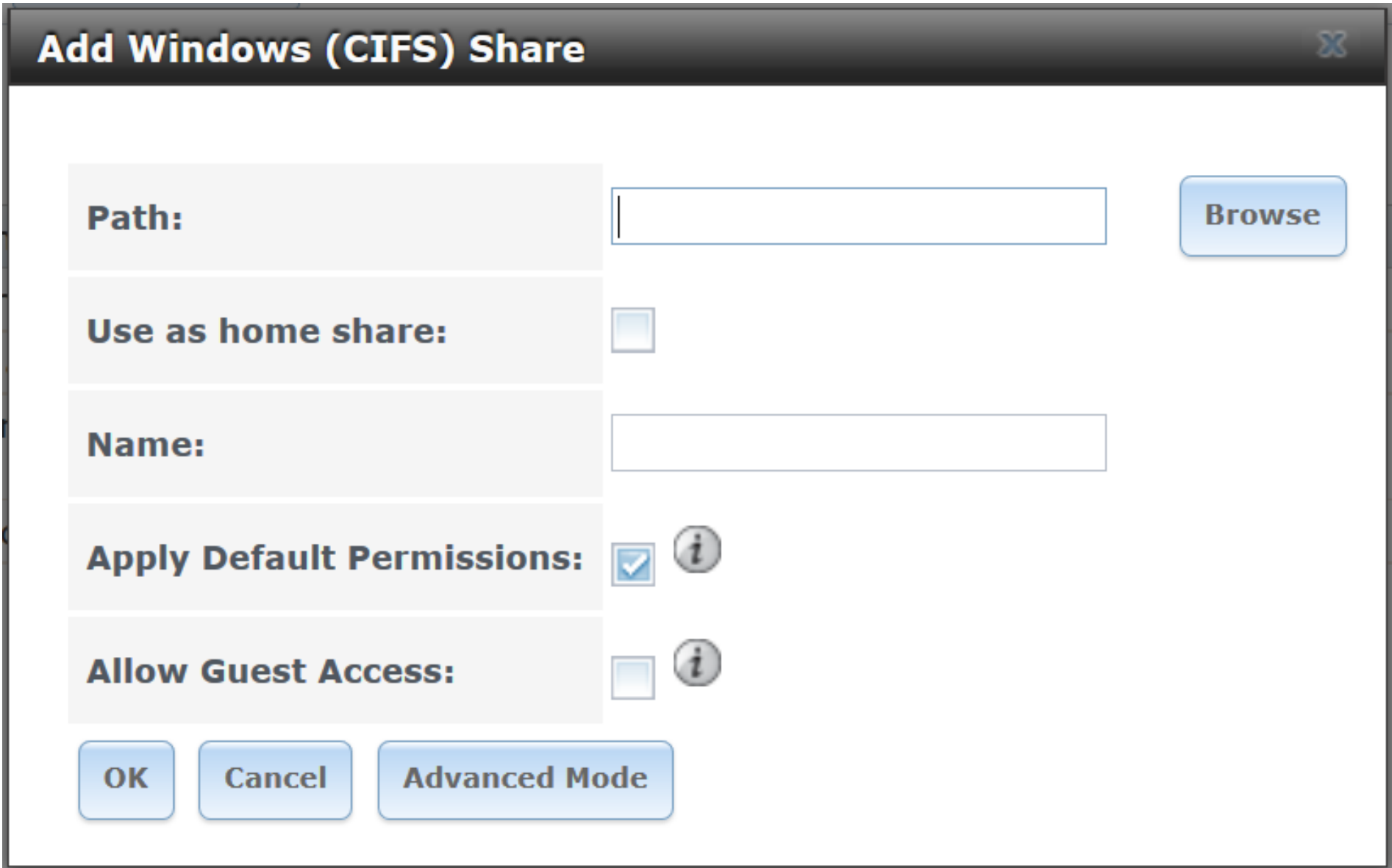


Figure 7: Adding a Share



Add Windows (CIFS) Share

Path: **Browse**

Use as home share: ☐

Name:

Apply Default Permissions: ☒ ⓘ

Allow Guest Access: ☐ ⓘ

OK **Cancel** **Advanced Mode**

Figure 8: Add Share menu (Windows/CIFS)

Each share type has an advanced mode for additional options. You will also need to make sure your share types are enabled in the Services menu (Fig 9). Each service has its own individual configuration menu, which you can read about in the FreeNAS documentation at https://doc.freenas.org/9.3/freenas_services.html.

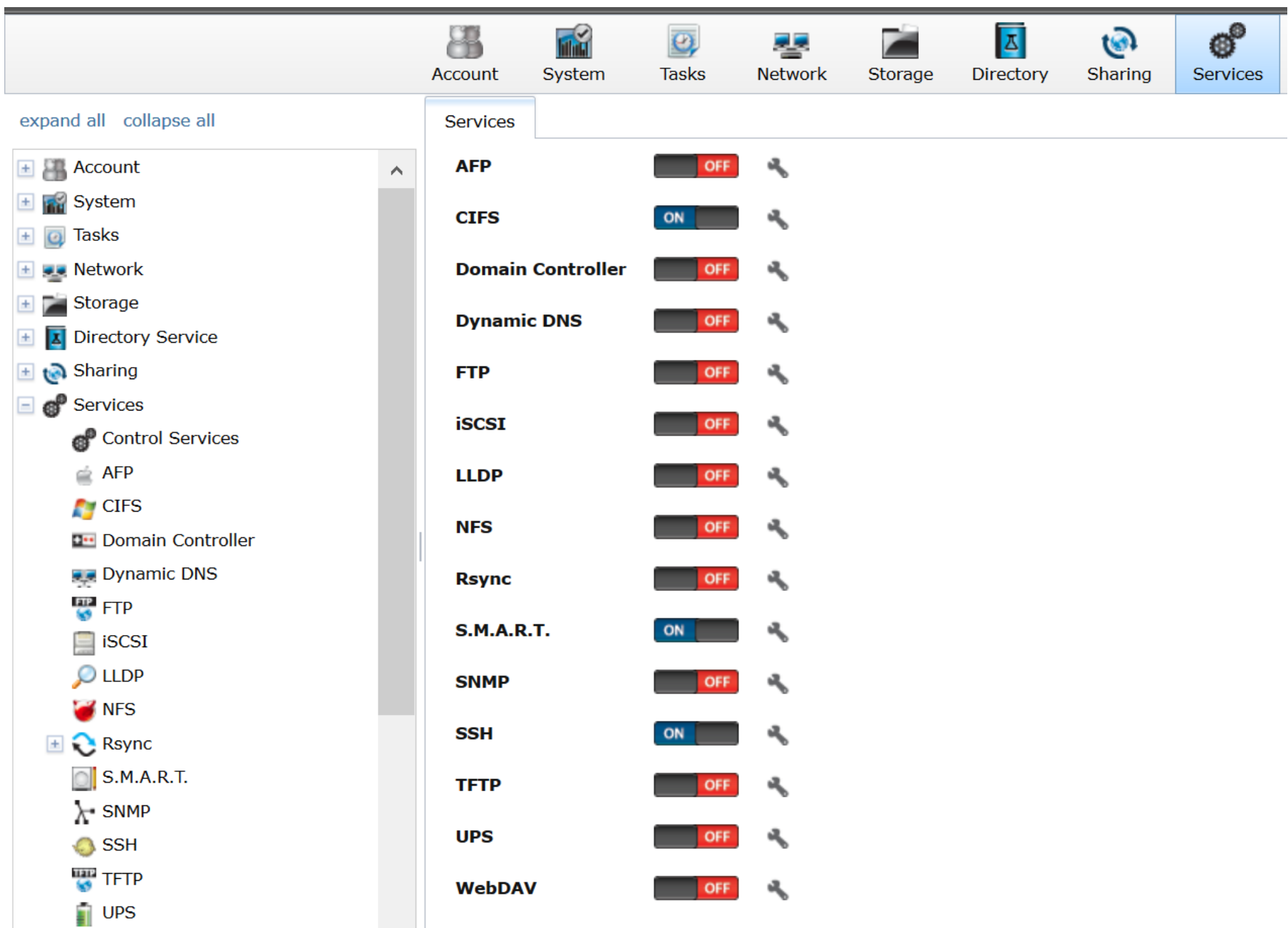


Figure 9: Enable your Share protocols in the Services Menu

Each share type has an advanced mode for additional options. You will also need to make sure your share types are enabled in the Services menu (Fig 9). Each service has its own individual configuration menu, which you can read about in the FreeNAS documentation at https://doc.freenas.org/9.3/freenas_services.html.

For full information on each type of share, please refer to the corresponding documentation:

Apple (AFP): https://doc.freenas.org/9.3/freenas_sharing.html#apple-afp-shares

Unix (NFS): https://doc.freenas.org/9.3/freenas_sharing.html#unix-nfs-shares

WebDAV: https://doc.freenas.org/9.3/freenas_sharing.html#webdav-shares

Windows (CIFS/SMB): https://doc.freenas.org/9.3/freenas_sharing.html#windows-cifs-shares

Block (iSCSI): https://doc.freenas.org/9.3/freenas_sharing.html#block-iscsi

The FreeNAS Team has also put together videos for setting up file based shares at https://www.youtube.com/watch?v=GVJQ0Vx_6i4&feature=youtu.be and block (iSCSI) shares at <https://youtu.be/HvyOWIFISdo>.

Snapshots

ZFS Snapshots are a great way to guard against lost data by saving your system state on a periodic basis without much processing overhead. Snapshots help to protect your storage against cryptolocker attacks, which you can read about on the iXsystems blog at <https://www.ixsystems.com/blog/defeating-cryptolocker/>. They help guard against human error, such as deleting the wrong file. They can also be helpful when you're upgrading to a newer version, especially if you're wanting to run off the nightly builds or alpha and beta versions. In addition, you can use them in conjunction with ZFS Replication to create data redundancy between multiple storage systems.

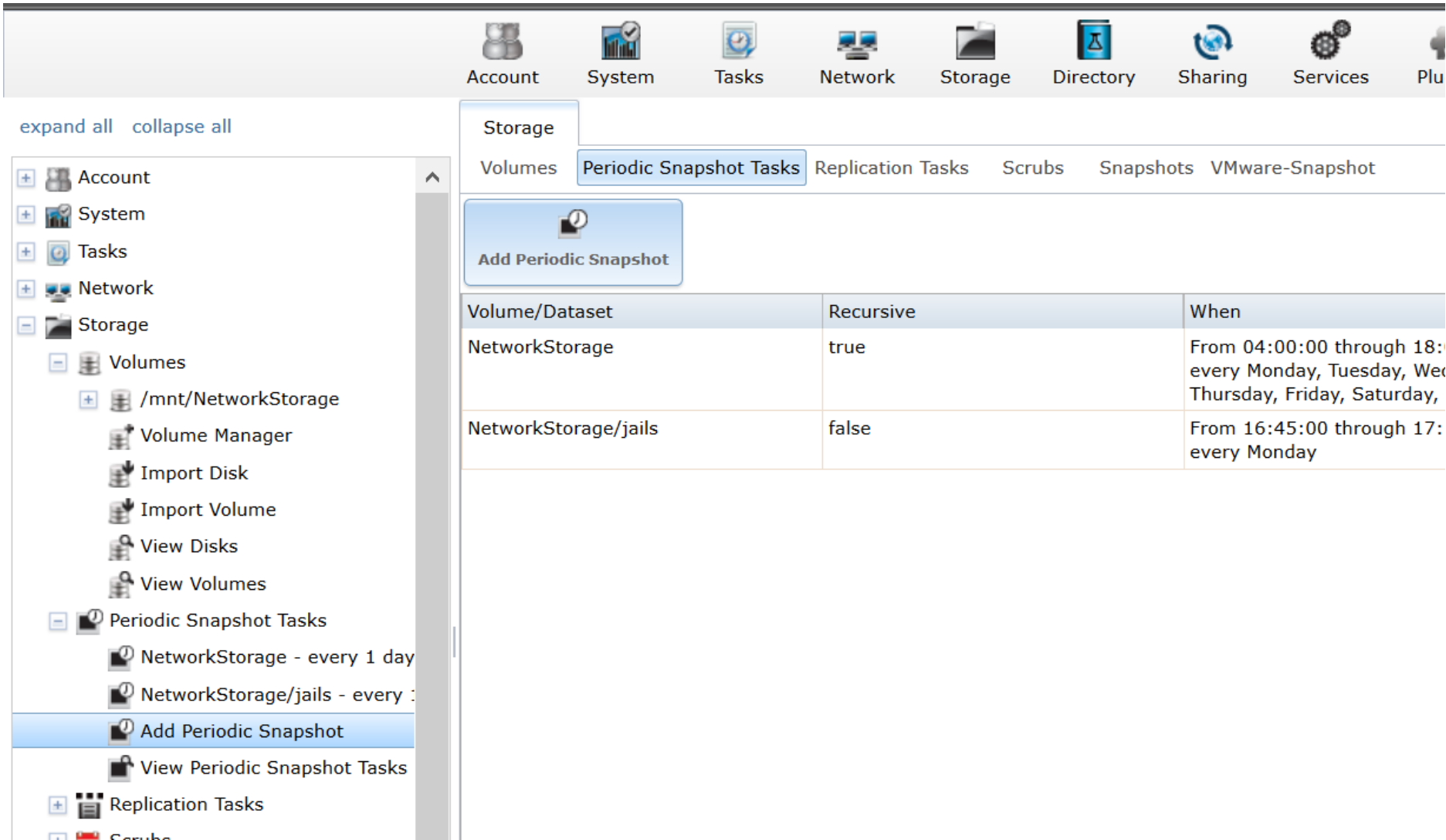


Figure 10: Adding a Periodic Snapshot Task

To set up a snapshot, simply go to your storage menu and go to your Periodic Snapshot Tasks section, then click on “Add Periodic Snapshot” (Fig 10). This will bring up a menu (Fig 11) that lets you set up your desired intervals between snapshot attempts, what days of the week you want them to occur and how long you want them to be saved on your system.

Periodic Snapshots

Volume/Dataset:

NetworkStorage

Recursive:

☐

Lifetime

2

Week(s)

Begin:

09:00:00

End:

18:00:00

Interval:

1 hour

Weekday:

☒

Monday

☒

Tuesday

☒

Wednesday

☒

Thursday

☒

Friday

☐

Saturday

☐

Sunday

Enabled:

☒

OK

Cancel

Figure 11: Snapshot Menu

For a full run down of adding a snapshot, you can take a look at the documentation at https://doc.freenas.org/9.3/freenas_storage.html?highlight=snapshot#periodic-snapshot-tasks.

FreeNAS Street

The FreeNAS team has also put together a video on the subject at <https://www.youtube.com/watch?v=yxnJH-8YvC8>.

Monitoring Disk Health with S.M.A.R.T.

Being able to monitor the health of your hard disks in a storage volume can be very helpful in preventing data loss. S.M.A.R.T. is a monitoring tool found on disk drives that reports on various aspects of drive health. Setting up S.M.A.R.T. in FreeNAS is very easy and can send email messages when issues arise. Just go to the S.M.A.R.T. section of your Tasks Menu and click on “Add S.M.A.R.T. Test” (Fig 12) to bring up the S.M.A.R.T. task menu (Fig 13).

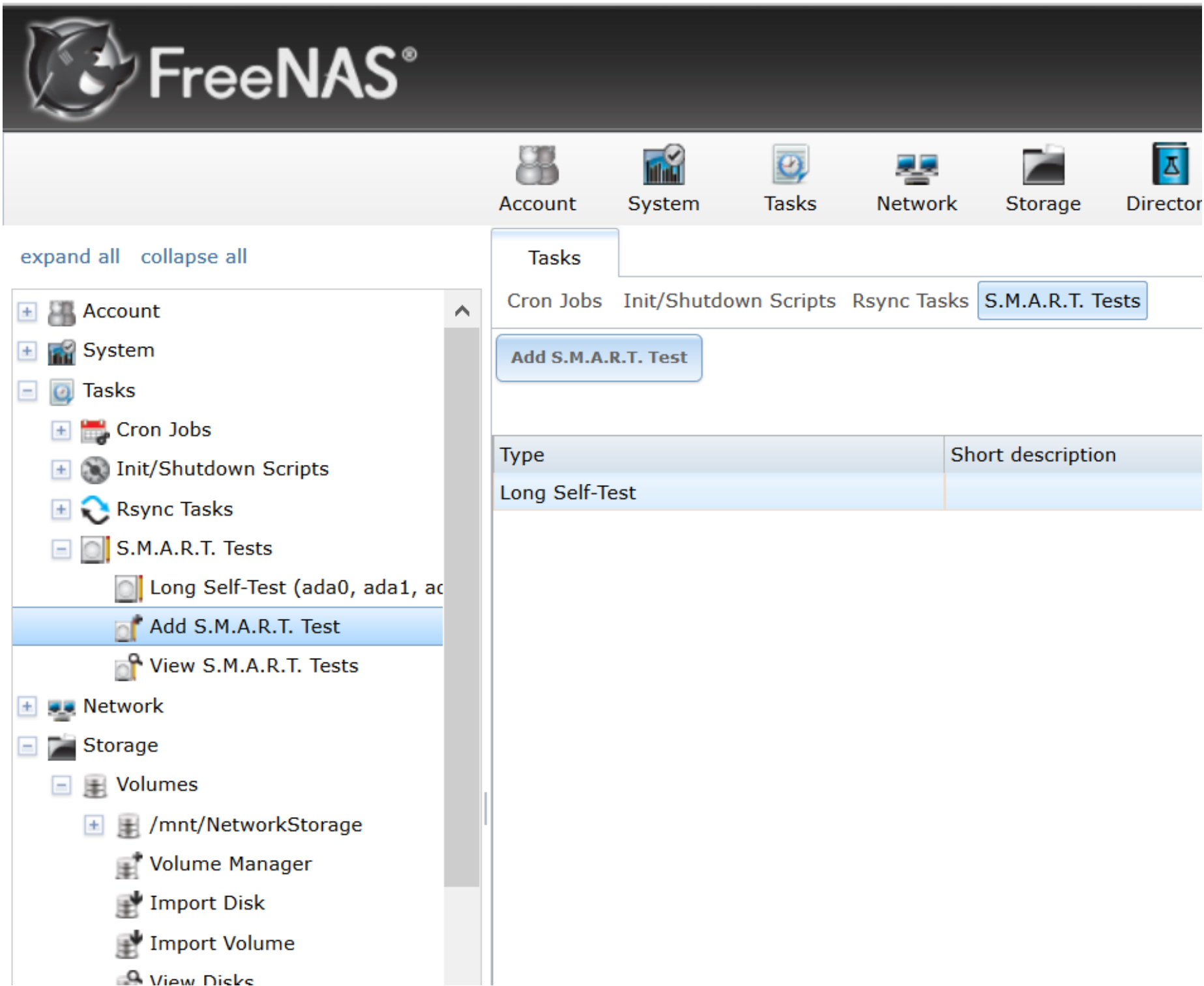


Figure 12: Adding a S.M.A.R.T. task

Once your menu is up, you can select the type of S.M.A.R.T. test you want along with when the test is performed. Click OK once you have everything set to your preferences.

Long Self-Test (ada0, ada1, ada2)

Disks:

ada0

ada1

ada2

Type:

Long Self-Test

Short description:

Hour:

Every N hour

Each selected hour

00

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Day of month:

Every N day of month

Each selected day of month

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

Figure 13: S.M.A.R.T. Task menu

FreeNAS Street

Once your S.M.A.R.T. task is enabled, go to the FreeNAS Services menu and make sure it is enabled. You can also set up additional settings, like temperature alerts and the email to send your S.M.A.R.T. notifications from the Settings menu (Fig 14) by clicking on the wrench icon.

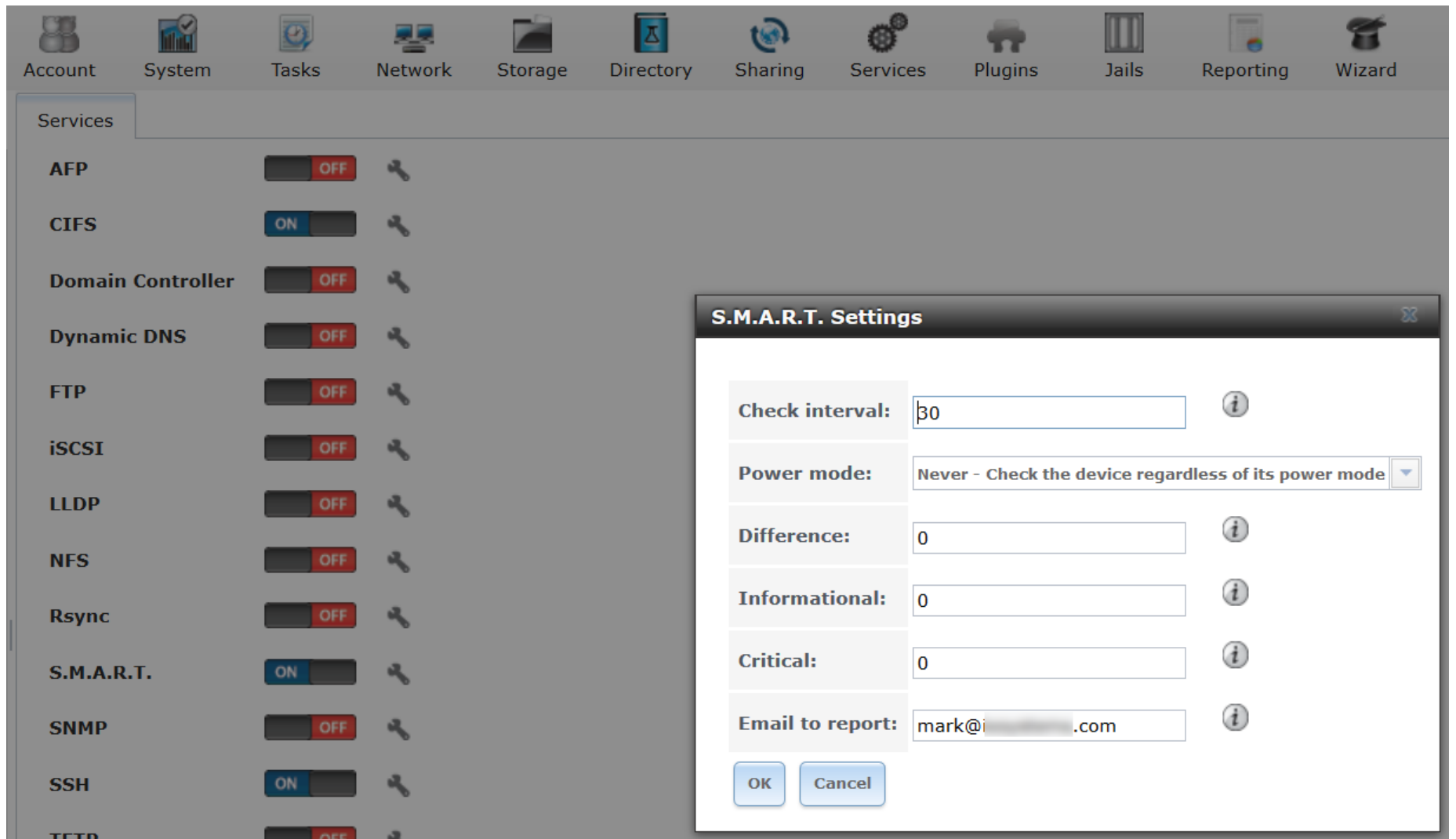


Figure 14: S.M.A.R.T. Service Settings Menu

You can find full documentation on S.M.A.R.T. tests at https://doc.freenas.org/9.3/freenas_tasks.html#s-m-a-r-t-tests.

Upgrading your FreeNAS Version

FreeNAS 9.3 and 9.10 both provide a simple way to upgrade to newer (or different) versions right from the User Interface, making manual downloads and installation unnecessary (though still possible). FreeNAS also saves boot environments to your OS drive to make switching back to previous versions as painless as possible.

When upgrading FreeNAS, it is first recommended to save your configuration. To do this, simply go to the “General” section of your System menu, then click on the “Save Config” button (Fig 15) at the bottom and save to a desired location on your computer.

FreeNAS Street

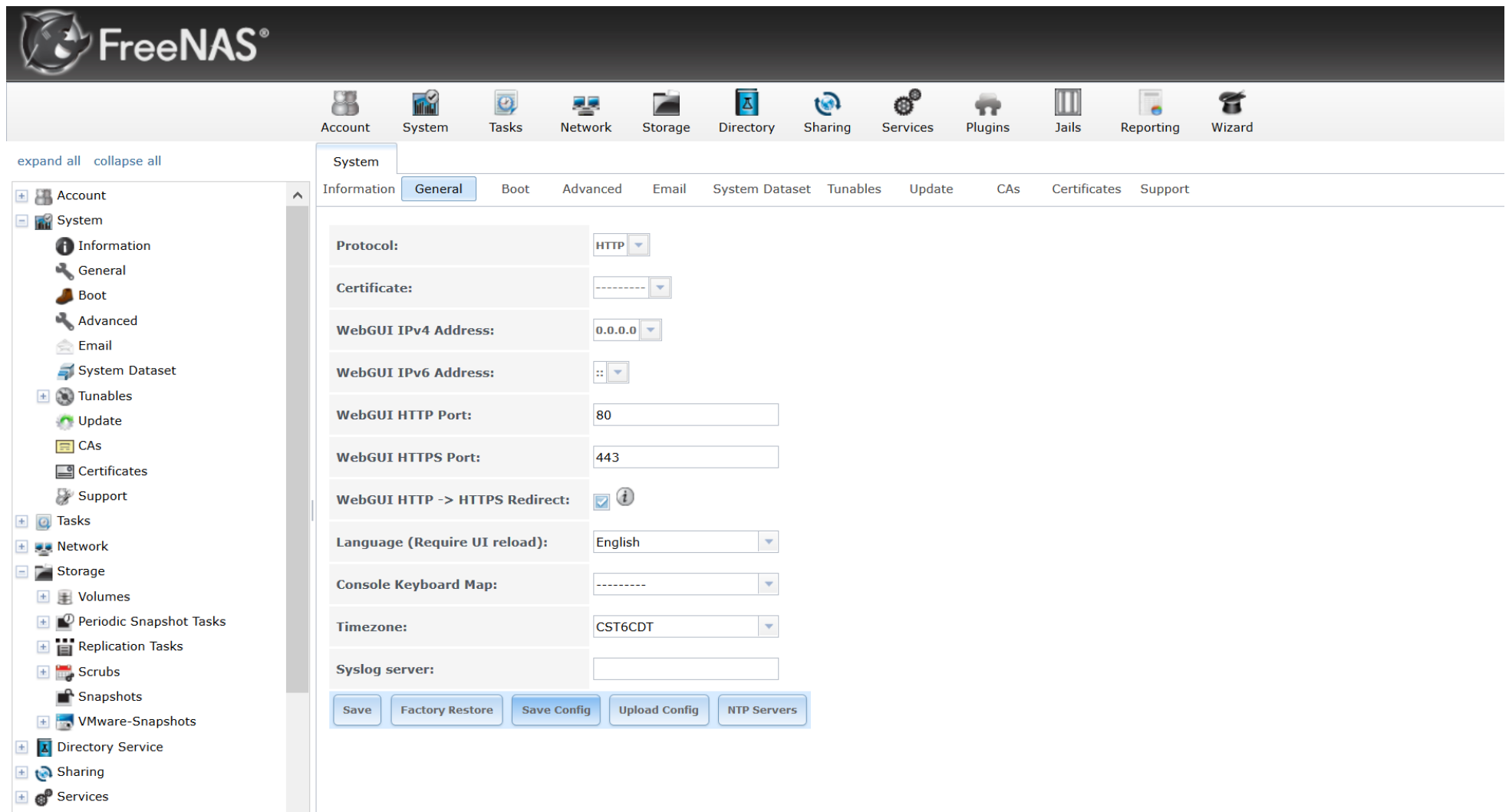


Figure 15: Saving your FreeNAS configuration

Once you have your configuration backed up, go to the “Update” section of your System menu. From there, click on the drop down menu on the right to select your desired FreeNAS version. Once that is selected, just click on the “Apply Pending Updates” button (Fig 16).

FreeNAS Street

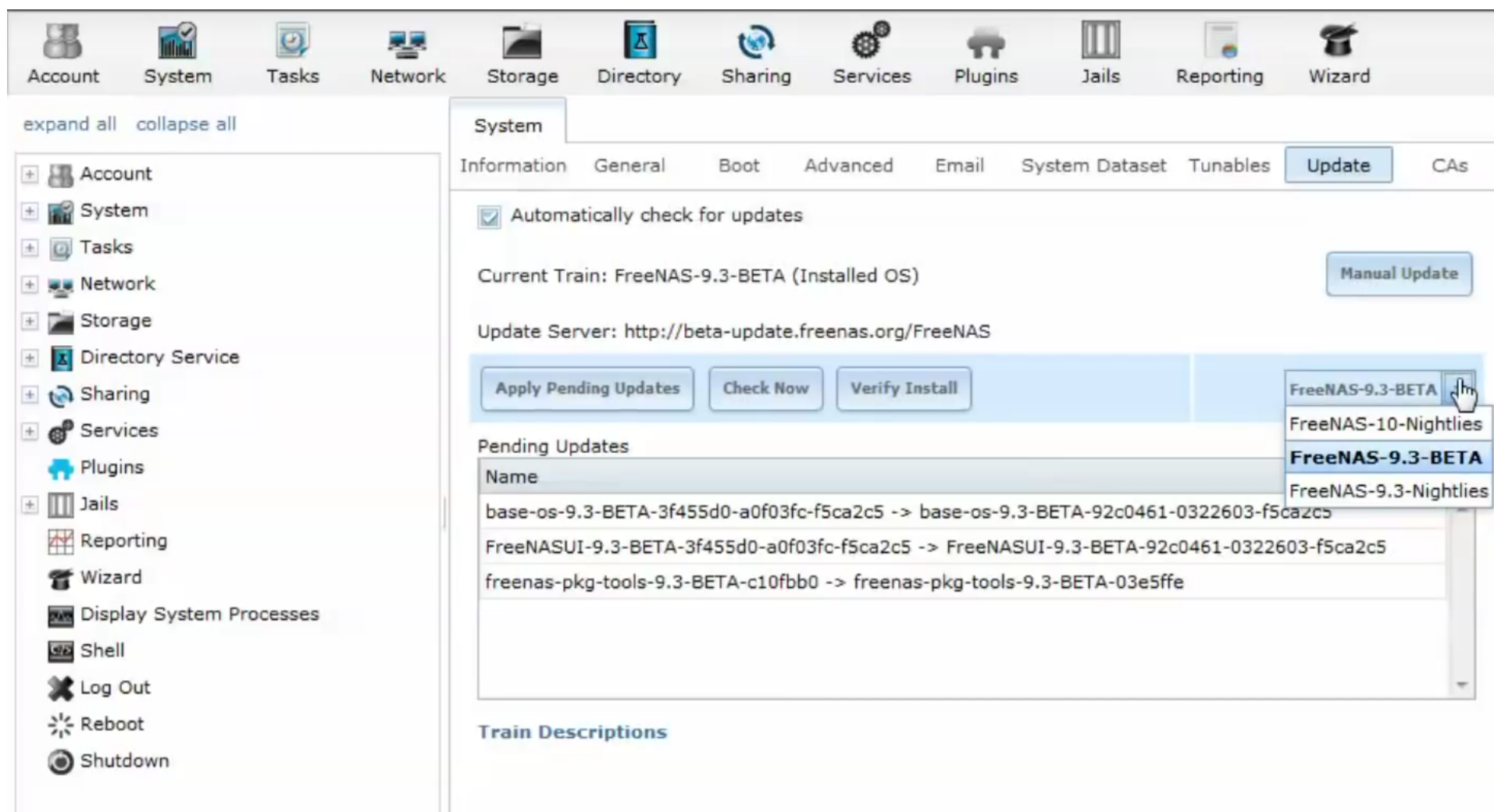


Figure 16: Upgrading your FreeNAS Version

Once your update downloads and installs, your system will automatically reboot.

For further information on updating FreeNAS, go to the FreeNAS Documentation at https://doc.freenas.org/9.3/freenas_install.html#upgrading-from-the-gui. The FreeNAS team has also put together videos for upgrading from various versions.

9.2: <https://www.youtube.com/watch?v=TJjUcuZjCps>

9.3: <https://www.youtube.com/watch?v=L61IJF98eP8>

9.10: <https://www.youtube.com/watch?v=2nvb90AhgL8>

Conclusion

This installment has covered most of the basic functions you'll need to set up and configure your FreeNAS storage manually from the user Interface itself. We plan on covering plug-ins and more advanced administration tasks in future installments. In the meantime, please check out the Additional Resources provided for more FreeNAS related guidance in the meantime.

Additional Resources:

Blogs:

- FreeNAS Best Practices: Part 1 | Part 2 | Part 3 | Part 4
- FreeNAS: A Worst Practices Guide

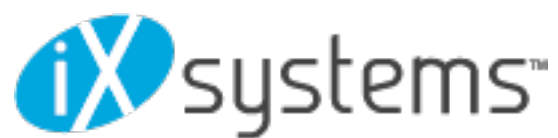
Forums: <https://forums.freenas.org/index.php>

Videos: <https://www.youtube.com/user/FreeNASTeam/videos>

FreeNAS Documentation: <https://doc.freenas.org/>

About the Author:

Mark VonFange has worked for iXsystems since 2008 in various roles including first response for professional services inquiries and developing marketing content. He has been an open source advocate for over a decade & enjoys building and repairing computers in his spare time.



USING FREEBSD AS A FILE SERVER WITH ZFS

In this course, we will learn how to use the current ZFS capabilities to help us build a home file server using FreeBSD 10.3.

Course launching date: 04th of July 2016

What will you learn?

- ZFS administration
- ZFS concepts and features

What skills will you gain?

- ZFS administration basics

What do you need?

- FreeBSD 10.3 with root privileges
- At least 10 GB free space

What should you know before they join?

- Basic FreeBSD administration knowledge



WORKSHOP

Module 1: FreeBSD and ZFS

Introduction to ZFS under FreeBSD

- Why ZFS on FreeBSD?
- ZFS features and concepts

Module 2 title: ZFS Administration

Module 2 description: Cover the commands and features to administrate ZFS volumes

- Create, destroy, list pools
- Zpools: single, mirrored, raid
- Understand ZFS properties

Module 3 title: Putting it all to work: Hosting our files using ZFS

Module 3 description: With the previous acquired knowledge, create a plan on how to organize our files and pools to host our files.

- Set ZFS properties based on the content of the files to host
- ZFS tuning
- Create a File Server using our pools

For more info visit our web page:

<https://bsdmag.org/course/using-freebsd-as-a-file-server-with-zfs-2/>

Don't hesitate to ask your questions at

marta.ziemianowicz@bsdmag.org



User Story From the OO Architecture Point of View

by Damian Czersnous

A good user story lays a great foundation for future work and shows engineering awareness of the team. For example, short sentences that follow deductive reasoning (top-down strategy) better corresponds with the way of ensuring object oriented architecture. How? In OO (Object Oriented) architecture, every method works in the context of its class. Every class works in the context of its package, and so on... The good OO architecture forms sentences starting from the top package to the bottom method.

Example of the good architecture:

```
com.sanecoders.bakery.productmgr.overview.ui.web.ProductOverviewPage.  
enter()
```

that translates to:

Sanecoders company owns bakery application that can present its products on the web page.

```
com.sanecoders.bakery.productmgr.editor.ui.web.ProductEditorPage.ente  
r()
```

that translates to:

Sanecoders company owns bakery application that allows to edit its product on the web page.

User story

This is an example of a user story. Let's find out how it helps organize OO architecture.

As a bakery customer, I would like to get familiar with available products to match them with my needs.

- Can I recognize what is what via drawings?
- Can I read how bakery products are made?
- Can I see a list of ingredients?
- Can I see all this on the homepage?

As an initial user story this can be fine. Although, an experienced Business Analyst or team might dig around „match them with my needs“. This may result in icons such as „eco“, „power“, „light“, „gluten-free“ or other that simplify searching.

Also „get familiar with“ should make professionals think, since customers do not always want to study products before buying even if they say so. Maybe presenting products that fit individual preferences in the first place would work better. In some businesses, software that learns users habits and makes decisions for them is seen as quite handy.

Imagine. You look for an eco rice bun, so you enter a bakery. You search for buns. Then, you read a list of ingredients and baking procedure to be sure it is eco (pure nature product that retains its characteristic after baking).

How much time does it take to find the eco rice bun? Wouldn't be nice to have the rice bun in the first place with „eco“ icon in a corner and already entered quantity? Or, how about your smart-phone that places an order of the two eco rice buns on Monday, and the three on Tuesday, because this is what you do anyway.

An experienced team asks the right questions, makes proposals, lets the customer feel the process he describes while defining the user story. Let's go back to the architecture.

User story - an intention sentence

The single sentence, „As a bakery customer..." presents well the idea of future functionality. It is easy to read and quick to understand. Great as a reminder. It describes customer intention without too many details. It also draws a foggy picture of the needs. These needs are explored with the „questions about needs“.

User story - questions about needs

Starting the questions with „Can I” helps to keep a client perspective. They should ask about effects and describe a friendly way to deliver the expected value. Their precision and accurateness reflect a quality of cooperation between a client and the team. They work in a context of the intention sentence and ensure correctness of the user story understanding.

The questions about needs can be quite helpful in ATTD (Acceptance Test Driven Development). It is easy to map them to the real test scenarios. They can be used as checkpoints during demonstration of the implemented functionality.

User story and engineering team

The questions about needs demonstrate team awareness of the functionality to be done. If the team doesn't understand functionality there is a problem with writing them. If the team questions such practice, especially in the complex systems, it usually means little experience.

The poor questions are the last warning for a leader and a product owner to start acting. The poor questions usually result with average effect (with small or medium mistakes) during presentation, often with an unfriendly user interface and architectural mistakes. Sometimes they expose a „don't care” attitude, which is a nightmare.

When OO architecture takes hits, it starts to generate serious maintenance costs. The main reason is that engineers don't really understand functionality and they use accidental words to describe architecture. Later on, they spend hours to find out how the code they write really works.

The best way to find out what engineers you have is to observe how they care about the user story: what questions they ask, what atmosphere they create cooperating with customer, how deeply they understand the impact of the right words on implementation.

User story flaws

Flaw 1. Long intention sentence and questions about needs

Long expressions reduce the speed of learning and the amount of acquired data. They cost more brain efforts to process them.

Example:

1. As a bakery customer, I would like to find a rice bun by looking at the drawings on the home page and be able to read a list of ingredients and baking procedure to match the bun with my healthy food preferences.

2. Can I recognize what type of products are available by looking at the drawings placed on the home page?

Make your sentences short and make a point with each. Spend some time constructing them. Don't make compromises. It becomes faster after a while.

Flaw 2. Duplicated information

The questions about needs work in the context of the intention sentence. There is no need to repeat yourself and make expressions longer. Working with the context or being constantly aware of the context leads also to a simpler design while coding.

Example:

Context: As a bakery customer, I would like to get familiar with available products to match them with my needs.

1. Can I recognize what is what among available products via drawings? Is equal to Can I recognize what is what via drawings?
2. Can I read how bakery products are made to find the ones that match my needs? Is equal to Can I read how bakery products are made?

Flaw 3. Generalizations within the questions about needs

Some level of generalization works well with the intention sentence, which sets the context for the entire user story. However, the goal for the questions about needs are to be specific as much as possible.

Example:

- Can I recognize what is what on a page?

The part „what is what” works in the context of „get familiar with available products”. „What is what” is equal to the available products. However, the way of recognition „what is what” is too general. The same applies to the place where products can be found. It would be better to ask:

- Can I recognize what is what via drawings?
- Can I see all this on the homepage?



About the author:

Damian Czernous

Reasoning about software architecture fascinates me for 10 years now.

Lead Engineering Coach at Nokia.

www.sanecoders.com, [@DamianCzernous](https://twitter.com/DamianCzernous)

With current advances in technology and systems, has the sector reached the point of consuming itself?

by Rob Somerville

Your columnist is not in the best frame of mind this month as the spectre of outsourcing and redundancy complete with the blunt scythe of incompetence and muddled corporate thinking has manifested itself in the corridors of my employment. After a typical seagull style management meeting (fly in, defecate on everyone and fly out again), the organisation is in a state of shock, with a number of staff in tears and the last dregs of goodwill flushed down the toilet with any foolish idealism that the powers that be have our best interests at heart. I suppose I should look on the bright side, we have at least evolved from a mushroom management style where everyone is kept in the dark and fed manure. Nevertheless, brutal though it is, having gone through the current process, I have a lot more support for the more ruthless American style of dismissing staff, giving them a redundancy cheque and escorting them out the door with the boxes of their possessions. Clarity will not prevail on the thinning of our current herd for some weeks, with consultations, clarifications, and staff applying for the new posts which were so vaguely defined in the 1cm thick proposal. At least under the former system, the sword of Damocles falls swiftly.

What was clear from a strategic perspective is that our organisation has consumed copious quantities of over-ripe fruit from the tree of the

shiny suit salesman, and while they may be drunk on the benefits to the balance sheet on the short term, the hangover and potential long term liver damage are not selling points that the salesman will admit to. Alcohol is addictive, and once introduced, unless controlled with an iron discipline, soon seeks to dominate. C'est la vie. My argument that vendors generally are there to make profit and take advantage of an organisation's weakness rather than deliver on long term vision – especially in the cut-throat IT sector – has fallen on deaf ears.

But this is only a very small battle in the larger war that is currently taking place in society. Technological advancements are even making traditional IT roles redundant, not just the drivers of the driver-less cars. Centralisation seems to be the current management focus, and to hell with issues such as data protection, geopolitical stability or international law. While I have no problem with private cloud based systems, once introduced into the corporate environment, there is no end to the potential pain if something goes wrong. Just to start with, you are buying a service, and that service may be loosely defined in terms of technical specification to both the vendor's and customer's benefit. A good example of this is the classic ISP's "Uptime guarantee". The 99% uptime is fine, but what

with those 3.65 days (even if the vendor keeps his promises) when the system goes down at year end (or whatever)? We end up with a giant finger pointing exercise, often with lawyers involved. Now I am not trying to imply that in house systems are any more or less reliable than what is available externally, but the problem is another layer of risk, management, communication and complexity. You might be lucky and your vendor is a good guy, but what if he is bought out by third party who is less proficient or changes the rules of the game? At least with your in-house staff you have a strong degree of control, and the added advantage they intimately know the culture and what your business depends on.

The more layers you have in a system, not only does the inherent complexity increase but the resultant risk of failure. But this risk is exacerbated when twinned with an inherent drive to be efficient. You cannot have 100% efficiency in any system as the randomness of the universe always intervenes. The difficult customer. The corrupt bit of data. A statistical error. Or just a man with a big digger. We can optimise, hit the peak in the bell curve, but we cannot make systems foolproof as a bigger fool will always come along. I have always argued that humans are essential to efficient functioning of technology, for it cannot speak with its own voice. Google does not have a conscience other than what has been programmed into it. Yet, this collusion between technology and those less understanding of what happens underneath the bonnet has driven a new industrial revolution, but with far greater consequences than the last. If we are to accept that that nuclear bomb was the pinnacle of the last age, where are we headed?

Mass unemployment in the West as robots take over the menial tasks? A surveillance state that eschews freedom of expression? Or worse still, a Matrix like society where we are there to provide fuel for an elite bunch of technocrats by our bodies themselves?

I used to be a great believer in the old adage "You want to make yourself redundant". In the face of the ethics of efficiency, and particularly the downward sloping face of the bell curve, I am becoming more reticent. I love efficiency. I also love good design, but these factors must integrate humanity, and the ultimate ethos that technology makes a great slave but a poor master. For if the roles are reversed in this scenario, rather than encompassing the ideal of freedom by having technology reduce the drudgery in our lives, we will become slaves either to technology or the system itself, refusing to accept the fallibilities in both. Outsourcing was the first episode in this painful lesson of the ruthlessness of the marketplace, and the technology coming on-stream now threatens not just jobs in the IT sector, but across disciplines previously untouched, such as journalism. If we continue to ignore these trends, I don't believe for a moment that we will inherit a land of milk and honey with leisure time galore as predicted by the futurists of the 1950's and 60's. Like the shipbuilders and miners of that age, I can only see one outcome – permanent redundancy.